

Quantum-Inspired Immune Clonal Algorithm for solving Large-Scale Capacitated Arc Routing Problems

Ronghua Shang^a, Bingqi Du^a, Kaiyun Dai^a, Licheng Jiao^a, Amir M. Ghalamzan E.^b, and Rustam Stolkin^b

(a. Key Laboratory of Intelligent Perception and Image Understanding of Ministry of Education of China, Xidian University, Xi'an, China, 710071; b. Extreme Robotics Lab, University of Birmingham, Edgbaston, Birmingham B15 2TT, United Kingdom)

Abstract—In this paper, we present an approach to Large-Scale CARP called Quantum-Inspired Immune Clonal Algorithm (QICA-CARP). This algorithm combines the feature of artificial immune system and quantum computation ground on the qubit and the quantum superposition. We call an antibody of population quantum bit encoding, in QICA-CARP. For this encoding, to control the population with a high probability evolution towards a good schema we use the information on the current optimal antibody. The mutation strategy of quantum rotation gate accelerates the convergence of the original clone operator. Moreover, quantum crossover operator enhances the exchange of information and increases the diversity of the population. Furthermore, it avoids falling into local optimum. We also use the repair operator to amend the infeasible solutions to ensure the diversity of solutions. This makes QICA-CARP approximating the optimal solution. We demonstrate the effectiveness of our approach by a set of experiments and by Comparing the results of our approach with ones obtained with the RDG-MAENS and RAM using different test sets. Experimental results show that QICA-CARP outperforms other algorithms in terms of convergence rate and the quality of the obtained solutions. Especially, QICA-CARP converges to a better lower bound at a faster rate illustrating that it is suitable for solving large-scale CARP.

Index Terms—Large-scale CARP, quantum rotation gate, quantum crossover operator, the repair operator.

I. INTRODUCTION

Capacitated Arc Routing Problem (CARP) has been used for many problems, including mail delivery, clearing snow from the street and school bus planning [1~2]. CARP computes a path corresponding with the minimum cost. For example, consider a team of agents/vehicles is dispatched to different destinations from to complete a task, including delivering goods or services as some desired objectives are obtained and some constraints are satisfied, such as:

1) Each agent starts from a dispatch point, delivers goods or

services and finally ends at the dispatch point again.

2) Each agent can only complete one task at a time.

3) The total demand of the must be equal or less than the sum of the agent's capacities, denoted by Q .

CARP aims at minimizing the total cost of the all agents under the constraints some of which mentioned above [3]. To the best of our knowledge, although CARP problems are mostly large-scale in practice, the approaches that are developed so far provide solutions only to the small-scale problems. Nonetheless, CARP is increasingly becoming more attractive for researchers because to find the optimal solution to the practical large-scale CARP problems is still a challenging open research question [4]. Many approaches to CARP problem have been developed, including heuristics and meta-heuristics. Heuristic algorithms obtain a higher quality solution using a smaller computational resource especially for small-scale problems as they have a simple algorithm structure. As an example, the solution of stereotype heuristic algorithm is used as an initial value in meta-heuristic approaches. This improves the convergence to an optimal solution and the stability of the solution. For instance, Golden et al. [5] proposed augment-merge heuristic algorithm that initialized the solution as many loops where each loop connects one task and warehouse. In the augment stage, the algorithm adjusts continuously assignment of the tasks in the loop to reduce the number of loops. Ulusoy heuristic algorithm proposed in [7] improves the path-scanning heuristic [6]. Augment-insert heuristic algorithm [8] and approximation algorithms [9] have been also proposed to compute an effective solution to the CARP problem. The stereotype heuristic CARP algorithm usually starts from a partial solution or an empty solution. It is then adjusted by a specific selection standard to obtain a complete solution. This type of algorithms converges to a sub-optimal solution very fast as its performance can be improved using intermediate rules. Despite these advantages, stereotype heuristic CARP is not a suitable approach to large-scale problems. On the other hand, the meta-heuristic algorithms, including Evolutionary Algorithms [10~11], Ant Colony System [12], Simulated

This work was partially supported by the National Natural Science Foundation of China, under Grant 61371201, the National Basic Research Program (973 Program) of China under Grant 2013CB329402.

Annealing, Tabu Search [13], Memetic Algorithms and so on [14~16], has been increasingly attracting researchers' attention because it has shown the performance of the approach to various types of combinatorial optimization problems with a wide range of applications. Using these algorithms, one or multiple solutions are obtained at each iteration which are used to generate the next generations. This procedure repeats resulting in an optimal solution.

There are also many meta-heuristic algorithms developed specially to solve CARP problems. For example, Hertz et al. [17] proposed tabu search algorithm in which the solution is represented by a full path. This approach obtains a solution with a higher quality than the one obtained by heuristic. More recently, Hertz and Mittaz [18] proposed an approach called Variable Neighborhood Descent (VND) to avoid a local optimum by defining several neighborhoods and making a detail exploration of the space in the neighborhood of the solution. Although this approach avoids a local optimum, it does not consider the infeasible solution space. Lacomme et al. [19] improved Genetic Algorithm (GA) and combined it with local search algorithm. The encoding method of MA is very practical and natural as two arcs in two opposite directions represent each demand side. Meta-heuristic algorithm has shown to obtain a high quality solution in most test cases, especially in those larger-scale instances, in the short period of time. In our works [20][21], we proposed immune clonal selection algorithm to solve multi-objective problems. This approach uses a clone operation to achieve a global optimum. Nevertheless, the clone operation increases population size in local space in order to improve the capability of local optimization. The immune clonal selection algorithm chooses portion of the antibody to mutate during the mutation operating to improve the diversity of population, and this operation often leads to the loss of the optimal solution. Memetic Algorithm with Extended Neighborhood Search (MAENS) proposed in [22] uses an extension step for local search. Hence, it rarely falls into a local optimum. Although this approach determines a good lower bound on the test case, it does not show a good stability property. Recently, according to MAENS, Mei et al. proposed an excellent cooperation co-evolutionary algorithm based on route distance grouping (RDG-MAENS) [23], which adopts the strategy of co-evolution [24] and divide and conquer [25]. They showed their approach outperforms other existing algorithms proposed to solve CARP. However, RDG-MAENS has the following shortcoming: (1) each generation includes some rounds to evolve, e.g. 50 rounds, and the best solution recognized so far is updated only after the evolution of one generation. Hence, the best solution recognized so far takes part in the evolution of the current round despite the fact that it may not be the best at the moment. (2) Assigning individuals to the appropriate subpopulation according to their distance can lead to an

imbalance number of assigned individuals to different sub-populations well as imbalance allocation of resources. Therefore, the solution obtained by this approach may neither be optimal nor stable.

In this paper, we propose Quantum-Inspired Immune Clonal Algorithm for solving Large-Scale CARP (QICA-CARP). First, we use heuristic algorithm to generate the initial solution to accelerate the convergence speed. Second, an antibody of population is used which is denoted by quantum bit encoding. In order to accelerate the convergence rate of the original clone operator, quantum rotation gate is used and the angle of rotation policy is dynamically adjusted to control the direction of the evolution of the population. Then, the quantum crossover operation is used to enhance information exchange while it increases the diversity of the population and achieves global search. We use also the quantum coherence to carry on a full interference crossover so that it overcomes the problem related to the shortcoming of early-maturing of individuals in the late time of evolution. Finally, the repair operator is used to improve the stability of the solutions as well as to maintain the diversity of the solutions to handle the infeasible solution set.

The rest of the paper is organized as follows. The description of CARP is introduced in Section II. In Section III, QICA-CARP for solving CARP is introduced. In Section IV, The experimental studies are carried out in several famous CARP test set (small scale, medium scale and large scale). Section V gives the conclusion and future work.

II. DESCRIPTIONS OF THE CARP

In [26], CARP is defined as a connected graph $G = (V, A, E)$, where V , A , and E denote a set of vertexes, arcs and edges, respectively. In a graph, each arc (or edge) corresponds with a non-negative cost and a non-negative demand. The demand $T \subseteq E \cup A$ is a set of tasks that vehicles are supposed to complete, e.g. delivering a service. Task $t \in T$ has two attributes, including demand for service, denoted by d where $d(t) > 0$, and service consumption, denoted by S where $S_C(t) > 0$. Non-mission side, i.e. an edge to which no task is assigned, does not need to service. If no service is provided from v_i to v_j , the consumption of the vehicle is $dc(v_i, v_j) > 0$. Here, we consider a connected graph to be symmetric, i.e. $dc(v_i, v_j) = dc(v_j, v_i)$ for each edge (v_i, v_j) . To find a plan of paths for m vehicle, whose capacities are Q , to serve all the tasks in the graph subject to some constraints is the main objective of solving a CARP problem. Eventually, a sequence of tasks can be expressed as a sequence of vertexes as the task sequence is supposed to connect the least-cost path between adjacent tasks.

In this paper, quantum-coding is used to initialize population as the code reflects the probability of the task being serviced in a sequence of vertexes. The solution x to CARP is expressed as a set of routes (m vehicles), as follows:

$$\min total_cost(s) = \sum_{q=1}^m cost(R_q) \quad (1)$$

$$cost(R_q) = \sum_{i=1}^{|R_q|-1} sc(v_{qi}, v_{q(i+1)}) \bullet w_{qi} + \sum_{i=1}^{|R_q|-1} tc(v_{qi}, v_{q(i+1)}) \bullet (1 - w_{qi}) \quad (2)$$

$$st: \sum_{q=1}^m \sum_{i=1}^{|R_q|-1} w_{qi} = M \quad (3)$$

$$(v_{qi}, v_{q(i+1)}) \in E \cup A, \forall w_{qi} = 1, 1 \leq h \leq m \quad (4)$$

$$\sum_{i=1}^{|R_q|-1} d(v_{hi}, v_{h(i+1)}) \bullet w_{qi} \leq Q, 1 \leq h \leq m \quad (5)$$

where, R_q is the q -th route of the routes set R , $|R_q|$ is the number of edges in q -th route, $sc(v_{qi}, v_{q(i+1)})$ is the cost of serving at edge $(v_{qi}, v_{q(i+1)})$. In addition, $tc(v_{qi}, v_{q(i+1)})$ is the travel cost of edge $(v_{qi}, v_{q(i+1)})$ which is calculated by the shortest path from v_{qi} to $v_{q(i+1)}$ obtained by Dijkstra algorithm in the connected graph [27]. If the i -th edge is served in q -th route and $w_{qi}=0$, otherwise. Constraints in eq. (3) and (4) ensure that each task is serviced by one vehicle whereas eq. (5) indicates the capacity constraint of the vehicles.

III. QUANTUM-INSPIRED IMMUNE CLONAL ALGORITHM FOR SOLVING CARP

Quantum-Inspired Immune Clonal Algorithm (QICA) is an immune clone algorithm based on quantum-coded, which applies quantum computing to clone operator of artificial immune system [20]. According to the quantum bit and the quantum superposition, we combine the feature of artificial immune system with quantum computation and use it to solve a CARP problem. In our approach, a quantum bit encoding represents an antibody of a population. The information of the current optimal antibody is then used to accelerate the convergence of the original clone operator and to control the population evolution, which evolves into a good schema with a high probability. This evolution occurs based on the action of quantum rotation gate as well as strategy of adjusting dynamically the angle of rotation. Furthermore, we adopt quantum crossover operation (1) to enhance information exchange (2) to increase the diversity of the population (3) to achieve a global search based on the characteristic of strong local optimization capability of cloning operator. Besides, the infeasible solutions are modified in the population so as to maintain the diversity of the population.

Compared to the traditional GA and MA (memetic algorithm) approach, QICA-CARP mainly includes the clone operation [28], mutation operation of quantum rotation gate as well as quantum crossover and repair operation, among which clonal operation enhances the local search [29~30]. If an individual has a higher affinity between antigen and antibody, the individual is of a larger clone scale. Hence, the algorithm with clonal operator can make a detailed search in the promoting space. The crossover is a widely. However, it is implemented between two individuals in GA and MA. This

results in a limited exchange of information. To make more individuals participate in the crossover process and to utilize the full knowledge of individuals, the special crossover method of quantum crossover can be used wherever (1) more than two individuals exist and (2) the quantum crossover happens in more than one position of an individual at the same time [30]. It also improves the premature convergence of individuals in the last stage of the algorithm by utilizing the characteristic of quantum coherence. After applying crossover, it may produce many infeasible solutions corresponding with cost that is lower than the others. We use a repair operator so that the infeasible solution can be considered to be feasible. This allows a detailed search resulting in a better solution. A scheme of QICA-CARP flow chart is shown in fig. 1.

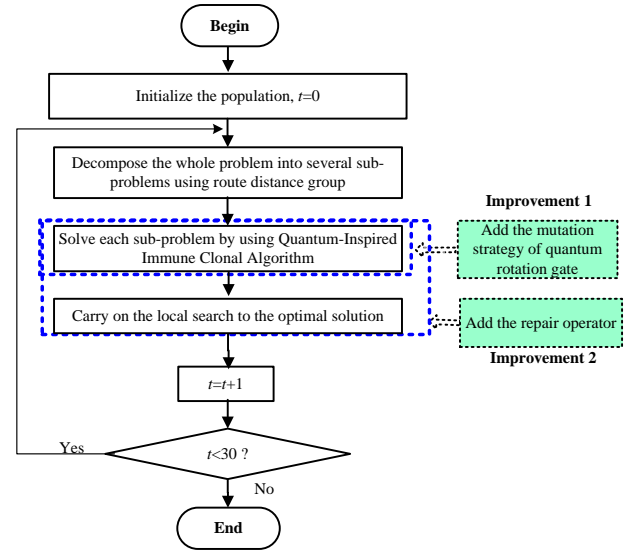


Figure 1 The flow of QICA-CARP

It has been proven that the improvement 1 in Figure 1 increases the speed of convergence while it handles the premature convergence effectively [31]. We refer to the repair operator as 3 main strategies to solve infeasible solution. In the first strategy, presented in [19], only the feasible solution is produced in assess process and any infeasible solutions are ignored. The second strategy is a common method of penalty function. The main disadvantage of this strategy is, however, a complex setting of parameters of the penalty function. Hence, it is very hard to keep the solution that has the best total cost and satisfies the constraint of capacity violation.

The third strategy designs a repair operator to repair the infeasible solution so that they are considered as a feasible one [32]. This is one of the very few methods for repairing the infeasible solution by adjusting a 0-1 vector in solution. However, it only considers a very limited area at each time in the solution space. In other words, it modifies one task in one route every time. Hence, it searches just the area around the

current solution instead of fully using the global information. The operator in the improvement stage takes global information into consideration to find the best solution of 0-1 vector combinations. In addition, it can find a better solution to improve the ability of searching. From the analysis that has just mentioned and comparing QICA-CARP with other EA, QICA-CARP and other EA all have the global search operator. However, the quantum crossover in QICA-CARP can make the best of individual's information and make a more effective global search to improve the searching ability of algorithm especially for the large scale CARP, so it plays a most important role in QICA-CARP. In fact, the repair operator is the main difference between QICA-CARP and other algorithms. It is also very helpful to look for a better solution by translating the infeasible solution, which is easily ignored and/or hard to handle, into a reasonable solution. This is of secondary importance. Finally, the mutation strategy accelerates the convergence speed while prevents premature convergence of individuals. It is, thus, an effective auxiliary operator in QICA-CARP.

A. Quantum bits and quantum coding

In QICA-CARP, we use a coding scheme based on quantum bits [30]. For quantum computing in an immune system, the smallest unit is a quantum information bit (a qubit), which is a two-state quantum system. Specifically, it is a unit vector defined in the two-dimensional complex vector space, and used to store information during the calculation process [30]. There are two basic forms of quantum information quantum bit at any instant, which are $|0\rangle$ and $|1\rangle$ respectively. Several bits can represent a task number, and a sequence of quantum bit can represent a task sequence that is transformed into a sequence of vertexes. We use QICA to solve CARP as follows. First, a population is initialized as $A(0)$ by quantum encoding. Quantum coding population of the t -th generation is then expressed as: $A(t) = \{a_1(t), a_2(t), \dots, a_n(t)\}$, where n represents the population size and $a_j(t)$ is a quantum chromosome. By contrast, $a_j(t)$ represents a path in CARP with the following expression:

$$a_j(t) = \begin{bmatrix} \alpha_{j1}(t) & \alpha_{j2}(t) & \dots & \alpha_{jm}(t) \\ \beta_{j1}(t) & \beta_{j2}(t) & \dots & \beta_{jm}(t) \end{bmatrix}, j = 1, 2, 3, \dots, n \quad (6)$$

where, m is the length of chromosome. $a_j(t)$ denotes a path that has m/n_{bit} tasks where n_{bit} is the number of bit representing a task number. Furthermore, the parameters in eq. (6) must satisfy normalization condition in eq. (7).

$$|\alpha_{ji}|^2 + |\beta_{ji}|^2 = 1, i = 1, 2, 3, \dots, m \quad (7)$$

where, α_{ji}^2 represents the probability of the i -th bit gene getting the state $|0\rangle$ and β_{ji}^2 is the probability of the i -th bit gene getting the state $|1\rangle$. The initialized population is denoted by $A(0) = \{a_1(0), a_2(0), \dots, a_n(0)\}$, initial iteration

denote by $t=0$, and quantum chromosome represents the path $a_j(t)|_{t=0} = a_j(0)$. In addition, the encoding of each antibody is

$$a_j(0) = \begin{bmatrix} \alpha_{j1}(0) & \alpha_{j2}(0) & \dots & \alpha_{jm}(0) \\ \beta_{j1}(0) & \beta_{j2}(0) & \dots & \beta_{jm}(0) \end{bmatrix} \quad (8)$$

A random number is then generated in the interval of $[0, 1]$ using the corresponding probability that the bit is 0 or the bit is 1. If the number is larger than α_{ji}^2 , the value of the bit is 1, and zero otherwise. After comparing m times, $A(t)$ is used to obtain a binary sequence which stands for the task number. The initialization of one solution is complete if all the task number is in the task sequence. An example of the encoding and decoding is described in Figure 2.

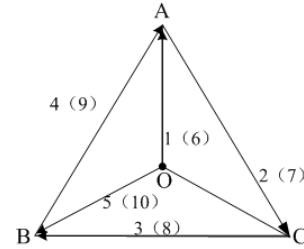


Figure 2 An example of encoding and decoding

In this figure, there are 5 tasks and 5 inverse tasks shown in parenthesis to serve 5 tasks no matter the task or its inverse task is served. We adopt 4 quantum bits to represent a task number where each bit has two attributes, including alpha and beta. After generating 4 random numbers and 4 times observations, it produces a sequence of 0 and 1 to represent a task number. As we know, 4 quantum bits can represent the non-negative natural number in the interval of $[0, 15]$. We use a modular arithmetic operation to link the number with the task number. First, we should initialize the quantum sequence $a(t)$, as follows:

$$a(t) = \begin{bmatrix} \alpha_1(t) & \alpha_2(t) & \dots & \alpha_4(t) \\ \beta_1(t) & \beta_2(t) & \dots & \beta_4(t) \end{bmatrix} = \begin{bmatrix} 1/\sqrt{2} & 1/\sqrt{2} & 1/\sqrt{2} & 1/\sqrt{2} \\ 1/\sqrt{2} & 1/\sqrt{2} & 1/\sqrt{2} & 1/\sqrt{2} \end{bmatrix} \quad (9)$$

For the first bit, we will produce a random number denoted by $rand$. If $rand$ is larger than $\alpha_1(t)^2$, the first bit of solution sequence gets 1 value, and 0 otherwise. After 4 such operation, four 0-1 bits are produced, which represents a task number. After producing 5 tasks number, a task sequence can be obtained. For example, although $\{1, 7, 3, 5, 9\}$ is a task solution, it is not reasonable because it does not satisfy the constraints of capacity violation.

We use Ulusoy's split method for decoding making the solution reasonable by considering the capacity violation constraint. As a sequence of tasks can expressed as a sequence of vertexes since the tasks can be linked by vertex, we can obtain a complete plan of the best necessary paths and their corresponding cost.

B. Immune Cloning Operation

One of the major steps of the Immune Clone Algorithm is Immune cloning operation. The clone operation is used to replicate a population. We consider the size of antibody population to be n and the antibody population to be represented by $A(t) = \{a_1(t), a_2(t), \dots, a_n(t)\}$. In addition we consider the antibody to be the candidate solutions to CARP. The affinity of the antibody and the antigen determines the value of the objective function at a candidate solution. Hence, the antigens are used to find the optimal solution and the antibodies represent the candidate solutions in the CARP. For a CARP, the function of this affinity for each quantum chromosome is:

$$f(A(t)) = \{f(a_1(t)), f(a_2(t)), \dots, f(a_n(t))\} \quad (10)$$

where, $f(a_j(t))$ represents the size of the affinity for the quantum chromosome $a_j(t)$. We select a population to be cloned for CARP, which is called dominant clonal population and is denoted by $A_s(t)$, based on the function of the affinity of the antibody and the antigen, i.e. the antibody-antigen affinity and the antibody-antibody affinity. The clone operation is then applied to $A_s(t)$ resulting in a cloned population, denoted by $A^{(1)}(t)$. The size of the antibody $a_j(t)$ after cloning operation is $q_j(t)$ where $q_j(t)$ is computed using eq. (11), as follows:

$$q_j(t) = \left\lfloor N_c * \frac{f(a_j(t))}{\sum_{i=1}^m f(a_i(t))} * \Theta_j \right\rfloor \quad j = 1, 2, 3, \dots, n \quad (11)$$

where $\lfloor \cdot \rfloor$ is an operation to round down the corresponding value. N_c relates to the size of the cloned population where $N_c > N_A$ and $N_A = |A_s(t)|$. Θ_j denotes the affinity between antibody j and other antibody. In the artificial immune system, this refers to the distance between the candidate solutions. In the case of a single antibody, the clone scale is adjusted according to the affinity of the antibody and the antigen as well as according to the affinity of the antibody and the antibody. The more the affinity of the antibody and the antigen stimulates, the greater the possibility that it is searched. In other words, if the affinity of the antibody and the antigen stimulates more, the scale of the clone will be larger. This is beneficial to finding the optimal solutions and maintaining the diversity of solutions to CARP. After applying the immune clone operation to the CARP, the obtained cloned population $A^{(1)}(t)$ will be as follows:

$$A^{(1)}(t) = \left\{ \{a_1^1(t), a_1^2(t), \dots, a_1^{q_1(t)}(t)\}, \dots, \{a_{|A_s(t)|}^1(t), a_{|A_s(t)|}^2(t), \dots, a_{|A_s(t)|}^{q_{|A_s(t)|}(t)}(t)\} \right\} \\ = \{a_1^{(1)}(t), a_2^{(1)}(t), \dots, a_N^{(1)}(t)\} \quad (12)$$

where, $a_i^k(t) = a_i(t)$, $k=1, 2, 3, \dots, q_j$. $a_i^{(1)}(t)$ expresses the cloned antibodies, $i=1, 2, 3, \dots, N$ and $N = q_1 + q_2 + \dots + q_{|A_s(t)|}$ which indicates the size of population after cloned operation. Thus, the immune cloning operation can be applied on the CARP following the steps above mentioned. It is worth mentioning

that Immune Clone operations clone antibodies with high adaptation resulting in more local search.

C. Variation of quantum rotation gate

The hyper-mutation in the immune clone operation is a kind of local search strategy. In fact, it is a random operation that only mutates part of the antibody to increase the diversity of population and it avoids falling into a local optimum. However, this operation usually reduces the diversity of the variation, which may lead us losing the optimal solution. Therefore, we present a variation of quantum rotation gate for solving this problem.

In the quantum theory, the migration between each qubit is realized through the quantum's transformation matrix [30]. The rotation angle of the quantum rotation gate is used to achieve the mutation operation and the population $A^{(2)}(t)$. This allows us to use the optimal path information totally in the variation. It also accelerates the speed of convergence [30]. The mutation operation of the side encoding is expressed in eq. (13).

$$\begin{bmatrix} \alpha'_{i\xi}(t) \\ \beta'_{i\xi}(t) \end{bmatrix} = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix} \times \begin{bmatrix} \alpha_{i\xi}(t) \\ \beta_{i\xi}(t) \end{bmatrix} \quad (13)$$

where, θ is the angle of the rotation mutation. After the mutation operation, the amplitude of probability of the ξ -th qubit in the quantum chromosome $a_i^{(1)}(t)$ can be expressed by eq. (13). In the algorithm, θ is equal to the adaptive dynamic step, denoted by δ . If the value of δ is too large, the search step will be consequently increased. This can easily lead to converging to a local extreme point which is called the precocious phenomenon. However, the optimal solution to CARP cannot be found by this approach. By contrast, if the value of δ is too small, the search step will be very small and convergence becomes very slow or even standstill. This is not practically helpful to solve the large-scale CARP. Hence, according to the characteristics of CARP, we define δ as a variable related to the clone scale as per eq. (14).

$$\delta = 10 \times \exp(-q_j(t)/N_c) \quad (14)$$

where, $q_j(t)$ is the size of the antibody $a_j(t)$ after cloning operation according to eq. (11), N_c is the total size of population after clonal operation. In addition, the search step can be adjusted adaptively based on the affinity of the antibody. After applying the variation of quantum rotation gate on the cloned population $A^{(1)}(t)$, the obtained population $A^{(2)}(t)$ can be represented as:

$$A^{(2)}(t) = \{a_1^{(2)}(t), a_2^{(2)}(t), \dots, a_N^{(2)}(t)\} \quad (15)$$

where, $a_i^{(2)}(t)$ represents the antibody $a_i^{(1)}(t)$ after the variation of quantum rotation gate. If ξ -th bit in $a_i^{(1)}(t)$ had a mutation operation like eq. (13), $a_i^{(2)}(t)$ after mutation can be computed by eq. (16).

$$a_i^{(2)}(t) = \begin{bmatrix} \alpha_{i1}(0) | \alpha_{i2}(0) | \dots | \alpha'_{i\xi}(0) | \alpha_{im}(0) \\ \beta_{i1}(0) | \beta_{i2}(0) | \dots | \beta'_{i\xi}(0) | \beta_{im}(0) \end{bmatrix} \quad (16)$$

We can also reduce the mutation step size around the winning antibody to improve the capability of local search. Furthermore, it increases the mutation step size around the inferior antibody to perform a full quick search based on the specific structure of cloning operator. We can get a better lower bound by this strategy. This algorithm used to solve CARP is reported in Table 1.

Table 1 The variation of quantum rotation gate used in solving CARP

Algorithm 1: the algorithmic process of the variation of quantum rotation gate used in solving CARP

Input: the antibody population $A^{(1)}(t)$ obtained after the clone operation;
Output: The population $A^{(2)}(t)$ obtained after the variation of quantum rotation gate;

Begin

1. **Do** operation on the cloned population $A^{(1)}(t)$, map the decision variable to the unit interval and obtain the population $A^{(1)}(t)$, which is an individual within the interval of $[0, 1]$;
2. Set the quantum rotation angle θ according to the size of populations $A^{(1)}(t)$ and $A^{(1)}(t)$;
3. Set the direction of rotation based on the calculated antibody affinity of the cloned population $A^{(1)}(t)$.
4. **If** $(A^{(1)}(t)(i,j) > 1 \parallel A^{(1)}(t)(i,j) < 0)$.
5. Modify the updated illegal Individual;
6. **end if**
7. Output the population $A^{(2)}(t)$;

End

D. Clonal selection operation

In the immune clonal algorithm, we usually use the immune genetic manipulation to retain the information of the parent population and to maintain the diversity property. The immune genetic manipulation T_g^C comprises the cloning recombination T_r^C and the cloning variation T_m^C . Let $A_g(t)$ denote the antibody population after the immune genetic manipulation. We can now write the relation between T_g^C , T_r^C and T_m^C as follows:

$$T_g^C(A_g(t)) = T_r^C(A^{(1)}(t)) + T_m^C(A^{(1)}(t)) \quad (17)$$

In QICA-CARP, however, the immune selection of the immune cloning operation is different than the selection operation in the traditional EA. Besides, the new population consists of excellent antibodies selected from the offspring according to the quantum rotation gate mutation strategy and the corresponding with the parent population, i.e. $A^{(3)}(t) = T_s^C(A(t) \cup A^{(2)}(t))$. We now set a maximum size of the population, denoted by N_{\max} , before the clonal selection operation. Then, we determine the size of population based on the maximum size after the clonal selection operation. If $N > N_{\max}$, we calculate the affinity function of each antibody and select the antibody with a relative high affinity function. Otherwise, the scale of antibody population remains the same. Eventually, the population $A^{(3)}(t)$ after the clonal selection operation can be expressed as:

$$A^{(3)}(t) = \{a_1^{(3)}(t), a_2^{(3)}(t), \dots, a_{N_s}^{(3)}(t)\} \quad (18)$$

s.t.: $N > N_{\max}$: $N_s = N_{\max}$, $i = 1, 2, \dots, N_{\max}$;
 $N \leq N_{\max}$: $N_s = N$, $i = 1, 2, \dots, N$;

where, $a_i^{(3)}(t)$ denotes the antibody after the clonal selection operation. To select the best individuals from the progeny after the clone proliferated and to form a new population of

antibodies is the main purpose of this operation. This increases a faster convergence to a better solution for CARP problem.

E. Quantum crossover

In the evolutionary algorithms, the crossover operation is another strategy to search for the optimal solution, such as single-point crossover, uniform crossover, multi-point crossover, and arithmetic crossover. To limit the two individuals is one of the common crossover operations. To prevent the solutions from falling into a local optimum, we use quantum coherence to construct the full interference crossover [30]. In this approach, more than two antibodies in the population are involved in the crossover for solving the CARP problem. Here, we use the full interference crossover for CARP. We select randomly four loops in the antibodies, which are represented by R_1, R_2, R_3 , and R_4 where, $R_1 = \{0, A(1), D(2), C(3), B(4), A(5), D(6), 0\}$, $R_2 = \{0, B(1), A(2), D(3), C(4), B(5), A(6), 0\}$, $R_3 = \{0, C(1), B(2), A(3), D(4), C(5), B(6), 0\}$, and $R_4 = \{0, D(1), C(2), B(3), A(4), D(5), C(6), 0\}$. In the connectivity graph, each antibody of these four loops are involved in the crossover operation according to the characteristics of the full interference crossover. The full interference crossover of the loop is shown in Figure 3.

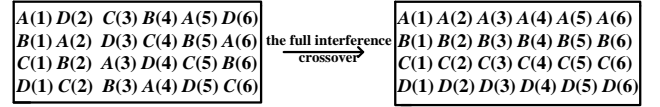


Figure 3 The schematic diagram of the full interference crossover

Fig. 3 shows the full interference cross, which is one kind of the cross-way by the diagonal re-assortment implemented on the solutions of the loop for CARP. We select $N_s/3$ individuals with a worse rank in population to perform crossover because it has a strong influence on the structure of the solution. The interference crossover process is as follows: (1) select several individuals and random positions, called pos , (2) each individual exchange bits with each other. The exchange is happening with 1 bit interval. The population $A^{(4)}(t)$ obtained after performing the full interference crossover for the population $A^{(3)}(t)$ can be written as per eq. (19), in the following:

$$A^{(4)}(t) = \{a_1^{(4)}(t), a_2^{(4)}(t), \dots, a_{N_s}^{(4)}(t)\} \quad (19)$$

where, $a_i^{(4)}(t)$ represents the antibody obtained by the full interference crossover operation. This quantum crossover method improves the one-sidedness and locality of common crossover by benefiting the most from the antibody information. Moreover, it can inject new impetus and produce new antibodies while the population evolves into the local optimal. This is very useful for solving a CARP problem. The full interference cross by using the properties of the quantum coherence allows the algorithm prevents the population from prematurely falling into a local optimum in the evolving process. However, the solution obtained after crossover may become infeasible because some tasks may be not included in

the solution. Hence, we adopt the repair operator to deal with the situation in the next section.

F. The repair operation

QICA-CARP is an evolutionary algorithm.

The initial population in EA is usually generated randomly resulting in a large number of infeasible solutions. Moreover, the offspring after QICA-CARP applied on the feasible solutions may also results in some infeasible solutions. Here, we use the heuristic algorithm that generates an initial solution speed up the convergence of the algorithm. However, this does not handle the infeasible solutions.

Let $\overline{A}^{(4)}(t) = \{\overline{a}_1^{(4)}(t), \overline{a}_2^{(4)}(t), \dots, \overline{a}_k^{(4)}(t)\}$ denote the infeasible solutions, then the population $A^{(4)}(t)$ obtained by the full interference cross is expressed as:

$$A^{(4)}(t) = \{a_1^{(4)}(t), a_2^{(4)}(t), \dots, a_{N-k}^{(4)}(t)\} \cup \{\overline{a}_1^{(4)}(t), \overline{a}_2^{(4)}(t), \dots, \overline{a}_k^{(4)}(t)\} \quad (20)$$

where k denotes the number of the infeasible solutions. The common strategies for processing the infeasible solutions includes multi-objective policy [26], repair infeasible solutions [33] and the penalty function method [34]. The penalty function method is a common method for solving constrained optimization problems. In this approach there exists a punishment for infeasible solutions to decreases the solution's fitness. The approach accepts non-feasible solutions so that some non-feasible solutions have the opportunity to be remained when it is appropriate; thus, some excellent solutions are retained expanding the search space. However, since the selection of the penalty coefficient and the penalty function is relatively complex, an appropriate function should be used with a good selection of the corresponding parameters according to the problem itself.

Considering the CARP to be an NP-Hard problem, it is very complex to use the penalty function to handle the infeasible solutions. We choose the appropriate repair operator to translate the infeasible solutions into the feasible solution of the population. A repair operator is designed, which can either convert the infeasible solutions into the feasible solutions or search in the convex border to find the optimal solution. Using the repair operator, population $A^{(5)}(t)$ can be computed as follows:

$$A^{(5)}(t) = \{a_1^{(4)}(t), a_2^{(4)}(t), \dots, a_{N-k}^{(4)}(t)\} \cup \{\overline{a}_1^{(4)}(t), \overline{a}_2^{(4)}(t), \dots, \overline{a}_k^{(4)}(t)\} \\ = \{a_1^{(5)}(t), a_2^{(5)}(t), \dots, a_N^{(5)}(t)\} \quad (21)$$

where, $\overline{A}^{(4)}(t) = \{\overline{a}_1^{(4)}(t), \overline{a}_2^{(4)}(t), \dots, \overline{a}_k^{(4)}(t)\}$ is the infeasible solutions after the repair operation and $A_i^{(5)}(t)$ is the antibody after the repair operation. The repair operator can be embedded into the optimization algorithms of CARP that improves the search efficiency. The flowchart of the CARP

algorithm with repair operator embedded is reported in Table 2.

Table 2

Algorithm 2: The repair operator algorithm embedded in CARP

Input: the infeasible solutions $\overline{A}^{(4)}(t)$;

Output: the feasible solutions $\overline{A}^{(4)}(t)$;

Begin

1. Calculate the affinity of the antibody in $A^{(4)}(t)$ denoted by *aff*;
2. For the repeated task in task sequence in $\overline{A}^{(4)}(t)$, we eliminate the one which make a higher travel cost;
3. **If** (*aff* < 1.3 * *Lower_Bound*)
4. **while** (*N* is not the empty set) **do** // Set *N* indicates all tasks which is not inserted in the loop
 5. For all tasks T_i in *N*, $1 \leq i \leq |m|$, look for the vehicle collection $\Phi(T_i)$ that consists of the vehicles which the task T_i can be putted in without capacity violation.
 6. Select the task which can make the $|\Phi(T_i)|$ (the number of elements in $\Phi(T_i)$) smallest to insert .
 7. Find the minimum current loading *TCLL* vehicle, then the task T_k can be assigned to the vehicle.
 8. Remove the task T_k from *N* and update the vehicle's loading.
- end while**
- else**
9. $i=1$;
10. Find the loop that does not meet the capacity constraints in the k -th vehicle loop, denoted by r . Then, find the total task *SUM* in the r loops.
11. The remaining $k-r$ loops are arranged in ascending order based on the loop requirements.
- while** ($SUM + f(\overline{a}_i^{(4)}(t)) > (i+r)*Q$) **do**
12. The loop $\overline{a}_i^{(4)}(t)$ is added to the task set *M*.
13. $SUM = SUM + f(\overline{a}_i^{(4)}(t))$.
14. $i++$;
- end while**
15. The loop $\overline{a}_i^{(4)}(t)$ is added to the task set *M*.
16. Do the path-scanning on all the tasks in the set *M* and use the Ulusoy-Split segmentation method optimization on it.
17. Use the new circuit to replace the old circuit in the $\overline{a}_i^{(4)}(t)$ to obtain $\overline{a}_i^{(4)}(t)$.
- end if**
- End**

Among them, $\overline{a}_i^{(4)}(t)$ ($1 \leq i \leq k$) indicates the loop formed by the i -th vehicle;

TCLL the current remaining capacity of j -th vehicle; *T* indicates the task that required to service; $|m|$ is the number of total tasks.

In summary, we adopt QICA-CARP to solve CARP whose corresponding steps includes quantum coding, immune cloning operation, variation of quantum rotation gate, clonal selection operation, quantum crossover and the repair operation. We initialize the quantum encoding population as $A(t)$. The state transfer of antibody population is denoted as follows:

$$A(t) \xrightarrow{\text{clone operation}} A^{(1)}(t) \xrightarrow{\text{quantum rotation gate}} A^{(2)}(t) \xrightarrow{\text{clone selection operation}} A^{(3)}(t) \xrightarrow{\text{quantum crossover}} A^{(4)}(t) \xrightarrow{\text{the repair operation}} A^{(5)}(t)$$

It is known that, the immune clone algorithm is an intelligent algorithm that has the strong capability of dealing with the local optimum by increasing the local population

size. However, this algorithm does not solve the large-scale problems and it fails to solve the real-time process. Hence, we use the parallel nature of quantum computation to solve CARP to improve the convergence rate of the original operator. Table 3 shows the algorithmic process of QICA-CARP.

Table 3 the algorithmic process of QICA-CARP

Algorithm 3: The algorithmic process of QICA-CARP

Input: A connected graph with weights G , task set Z , the capacity of vehicle Q ;

Output: The optimal feasible solution $\bar{s}(Z)$;

Begin

1. Initialize population $p(Z)$;
2. $\bar{s}(Z) = \arg \min_{s(Z) \in p(Z)} (tc(s(Z)))$;
3. Decompose the task set $Z = (Z_1, \dots, Z_g)$;
4. $j=1$;
5. Use the heuristic algorithm to construct a better feasible solution of Z_j ;
6. Set the termination condition and the initialization iterations;

While ($Indivi_Best \neq Lower_Bound$) **do**

7. $t=0$, initialize the quantum encoding population $A(0)$ and calculate the affinity of the antibody;
8. Obtain $A^{(1)}(t)$ by the cloning operation;
9. Perform the variation of the quantum rotation gate generating $A^{(2)}(t)$;
10. Perform the immune selection operation to form $A^{(3)}(t)$;
11. Calculate the affinity function of $A^{(3)}(t)$ and perform a quantum crossover operation to get $A^{(4)}(t)$;
12. The repair operator is used to repair the infeasible solutions in $A^{(4)}(t)$ obtaining $A^{(5)}(t)$;
13. $t=t+1$;
14. If the termination condition is not satisfied, go to step 5;

end while

15. Obtain the optimal feasible solution of sub-component $\bar{s}(Z_i)$;
16. until $j = g$
17. $\bar{s}'(Z) = \{\bar{s}(s(Z_1)), \dots, \bar{s}(s(Z_g))\}$;

If ($tc(\bar{s}'(Z)) < tc(\bar{s}(Z))$)

18. Update the optimal feasible solution of Z : $\bar{s}(Z) = \bar{s}'(Z)$;

end if

19. return $\bar{s}(Z)$;

End

Among them, *Lower_Bound* is the lower bound of the test case.

IV. SIMULATION RESULTS AND THE ANALYSIS

A. Experimental setting

To demonstrate In order to evaluate the performance of the proposed approach QICA-CARP, we perform some experiments and compare the results with the ones obtained by MAENS and RDG-MAENS. MAENS has been proven to be a sufficient algorithm for small- and medium-scale problems and RDG-MAENS has been proven superior to some representative algorithms for large-scale CARP (LSCARP).

The experimental test sets reported in [23] includes *gdb*, *val*, *Beullen'C*, *D*, *E*, *F*, *egl* and *EGL-G*. The *gdb* test set consists of 23 small-scale or medium-scale test instances. The instance set of the *val* includes 34 test instances generated by 10 different graphs. Next, the *Beullen'C*, *D*, *E*, *F* is constructed according to the transport network in the Flanders district of Belgium by Beullens. Each test set includes 25 test instances. The instances in test sets of *C* and *D* are generated

by from the same graph. Likewise, the instances in test sets of *D* and *F* are generated by from the same graph. The capacity of vehicles in instances of *D* and *F* are twice the capacity in the instances of *C* and *E*. The next data set is *egl*. The corresponding test set is generated based on the application problems of sprinkling gravels on roads to avoid road icing on the winter of the Lancashire in the United Kingdom. This test set consists of 25 instances based on two graphs by setting different capacity constraints and task sets for different test instance. Finally, 10 large-scale instances make up the *EGL-G* test set that includes 255 vertex and 375 edges, and different instances are associated with different task sets and capacities of vehicles. Each instance in each test set runs independently 30 times. Some parameters are used in this experiment are the number of sub-problems $\partial=2$, which is the same as RDG-MAENS, the size of the parent antibody population $P_{size}=30$, which is identical to the size of the angle of rotation θ and is computed by eq. (14). The main parameters are the number of generation, the ratio of clone q , the probability of mutation P_{ls} . In the process of decomposition, there is also the degree of fuzziness parameter β . Here, to confirm the sensitivity of these 4 parameters, we set several groups experiments on test instance *egl-s2-B* and we analyze the mean cost in 10 independent runs as the index.

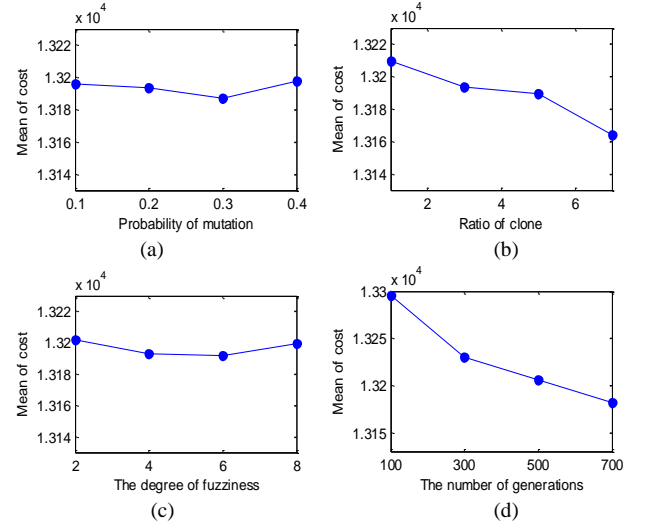


Figure 4 The sensitivity of the cost value to the parameters

It is shown in Fig. 4(a) the algorithm is not sensitive to the probability of mutation. If the probability of mutation is less than 0.3, the performance is getting better because the probability of mutation becomes larger. On the other hand, when it becomes larger than 0.3, the algorithm is getting worse. We empirically set the mutation value equal to 0.2. The mutation value less than 0.3 was shown to be good according to our experiments. As shown in Fig. 4(b), the output of the algorithm is sensitive to the clonal ratio. In fact, the larger the clone ratio, the less the cost. However, the

high ratio results in a high computing cost. Hence, a good balance between these two factors is very critical. We empirically set the clonal ratio to be 3 that has shown to have a good performance. As shown in Fig. 4(c), the output of the algorithm is not very sensitive to the degree of fuzziness. We experienced a good performance by the value of degree of fuzziness between 4 and 6, and we set it here equal to 5. On the other hand, the algorithm is very sensitive to the number of generation, as shown in Fig. 4(d). In fact, the larger the number of generations, the better the performance. We empirically set the number of generations equal to 500.

B. Comparing QICA-CARP with RDG-MAENS for solving CARP

We report the results achieved by QICA-CARP, RDG-MAENS and MAENS for solving the LSCARP with small-scale dataset in Table 4. Before results analysis we need to note the followings. (1) $|V|$, $|E|$, $|T|$ and LB represent the number of vertexes, the total number of edges, the number of tasks and the best lower bound found so far for the instances, respectively. (2) ‘mean’ represents the average value obtained by 30 independent runs on the instances. ‘Best’ represents the best one of them, and S_D is the standard deviation showing the stability of the algorithm on the test case. The performance of the algorithm is stable if the S_D is small. (3) The best results obtained by RDG-MAENS and QICA-CARP are in reported by bold fonts. In addition, the bold fonts report the same results. The best results obtained by MAENS or QICA-CARP is indicated by ‘_’. (4) ‘*’ indicates that the algorithm can get better lower bound than LB on the test case. (5) p and h are the parameters in the Wilcoxon signed rank test between QICA-CARP and RDG-MAENS. p represents the probability of the median of

two samples to be equal, and the null hypothesis should be questioned when p is close to 0. h is the test result where $h=0$ indicates that the difference between the median of the two samples is not significant whereas $h=1$ means the difference between the median of the two samples is significant. In this paper, the significance level of Wilcoxon signed rank test sets equal to 0.05. (6) We use percentage ‘%’ to show how much QICA-CARP makes the results better than the ones obtained by RDG-MAENS. This is denoted by Δ .

QICA-CAR and MAENS has almost the same optimal solution. However, The result obtained by MEAN has slightly better average value because gdb is small-scale data set and decomposition does not suit it. It is also evidenced that QICA-CARP achieves 16 smaller average value of the total_cost in the 23 test instances of gdb where other values are the same. Furthermore, S_D values of each test instance show that the standard deviation of QICA-CARP is 0 on 17 instances. Smaller standard deviations are obtained on other test instances in comparison with the results obtained by RDG-MAENS. This indicates the stability of QICA-CARP is stronger than RDG-MAENS on this test set. We see that QICA-CARP can converge to the lower bound of the test instance on all 23 instances of gdb by looking at the best values obtained by the algorithms on the test instances. The results of Wilcoxon signed rank test on this test set show that there are 14 “ $h=1$ ” in 23 test instances. This indicates that the results obtained by the new algorithm are better than the one obtained by the RDG-MAENS. This illustrates that QICA-CARP is much more effective than RDG-MAENS for searching the solutions on the gdb test set. However, because gdb is a small-scale dataset, we need to show the superiority of QICA-CARP with other datasets, as well.

Table 4 The comparison results of QICA-CARP and RDG-MAENS on gdb

name	$ V $	$ T $	$ E $	LB	MAENS			RDG-MAENS			QICA-CARP			p	h	$\Delta(\%)$
					mean	S_D	Best	mean	S_D	Best	mean	S_D	Best			
<i>gdb1</i>	12	22	22	316	316.0	0.0	316	316.0	0.0	316	316.0	0.0	316	1	0	0
<i>gdb2</i>	12	26	26	339	339.0	0.0	339	340.8	2.8	339	339.0	0.0	339	0.0039	1	0
<i>gdb3</i>	12	22	22	275	275.0	0.0	275	277.0	3.7	275	275.0	0.0	275	0.0039	1	0
<i>gdb4</i>	11	19	19	287	287.0	0.0	287	287.5	1.4	287	287.0	0.0	287	0.2500	0	0
<i>gdb5</i>	13	26	26	377	377.0	0.0	377	379.8	4.1	377	377.0	0.0	377	4.8828e-004	1	0
<i>gdb6</i>	12	22	22	298	<u>298.0</u>	0.0	298	299.1	4.2	298	298.3	1.3	298	0.6250	0	0
<i>gdb7</i>	12	22	22	325	325.0	0.0	325	325.0	0.0	325	325.0	0.0	325	1	0	0
<i>gdb8</i>	27	46	46	348	348.7	1.0	348	348.0	0.0	348	<u>348.2</u>	0.8	348	0.2500	0	0
<i>gdb9</i>	27	51	51	303	<u>303.0</u>	0.0	303	307.7	3.0	303	303.2	0.9	303	2.2442e-005	1	0
<i>gdb10</i>	12	25	25	275	275.0	0.0	275	277.4	4.0	275	275.0	0.0	275	0.0039	1	0
<i>gdb11</i>	22	45	45	395	395.0	0.0	395	396.5	3.4	395	395.0	0.0	395	0.0156	1	0
<i>gdb12</i>	13	23	23	457	<u>458.0</u>	0.0	458	461.1	6.0	458	458.4	1.3	458	1.2207e-004	1	0
<i>gdb13</i>	10	28	28	534	<u>536.0</u>	0.0	536	544.7	5.2	536	536.2	1.1	536	5.4777e-007	1	0
<i>gdb14</i>	7	21	21	100	100.0	0.0	100	100.4	0.8	100	100.0	0.0	100	0.0313	1	0
<i>gdb15</i>	7	21	21	58	58.0	0.0	58	58.0	0.0	58	58.0	0.0	58	1	0	0
<i>gdb16</i>	8	28	28	127	127.0	0.0	127	127.4	0.8	127	127.0	0.0	127	0.0313	1	0
<i>gdb17</i>	8	28	28	91	91.0	0.0	91	91.0	0.0	91	91.0	0.0	91	1	0	0
<i>gdb18</i>	9	36	36	164	164.0	0.0	164	164.0	0.0	164	164.0	0.0	164	1	0	0
<i>gdb19</i>	8	11	11	55	55.0	0.0	55	55.0	0.0	55	55.0	0.0	55	1	0	0
<i>gdb20</i>	11	22	22	121	121.0	0.0	121	122.7	0.7	121	121.0	0.0	121	3.4142e-007	1	0
<i>gdb21</i>	11	33	33	156	156.0	0.0	156	156.7	1.0	156	156.0	0.0	156	9.7656e-004	1	0
<i>gdb22</i>	11	44	44	200	200.0	0.0	200	200.4	0.8	200	200.0	0.0	200	0.0313	1	0
<i>gdb23</i>	11	55	55	233	<u>233.0</u>	0.0	233	234.9	0.4	233	233.4	1.4	233	1.5188e-004	1	0

Fig. 5 shows the convergence of some instances that illustrates more detailed description of comparison between the results obtained by QICA-CARP and RDG-MAENS applied on the *gdb* dataset. The x-axis represents the computational time in seconds, and the y-axis represents the average total cost of the best-so-far solutions over the 30 independent runs. The blue line represents the result of RDG-MAENS and the red line is QICA-CARP.

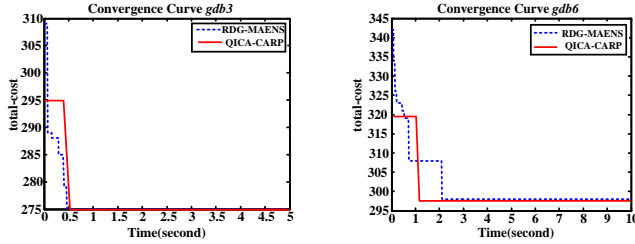


Figure 5 The convergence curves of QICA-CARP on some instances of *gdb*

It is evidenced in this figure that the convergence rate of QICA-CARP is slower than RDG-MAENS on some test set. However, QICA-CARP can converge to the lower bound with better stability property. Hence, the convergence rate of QICA-CARP in some instances of the small-scale test set *gdb* needs to be improved.

To compare the statistical properties of QICA-CARP with RDG-MAENS on the test set *gdb*, use the box diagram about 'total_cost' on each instance. Fig. 6 shows the box diagram of results obtained by applying QICA-CARP and RDG-MAENS on some instances of the *gdb* test set.

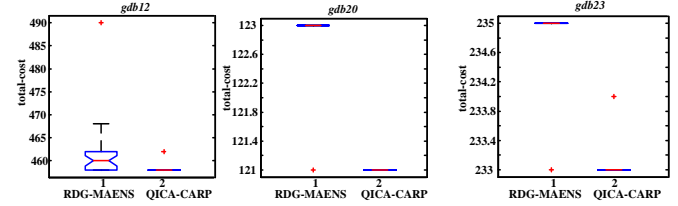


Figure 6 The box diagram of QICA-CARP and RDG-MAENS on *gdb*

The maximum value, the median line and the data distribution in Fig. 6 show that the solution obtained by QICA-CARP is much better than the ones obtained by RDG-MAENS. Besides, QICA-CARP can find a better optimal solution, which is consistent with the test results of these two algorithms in table 4. Because the *gdb* is a small-scale test set, the searching ability of these two algorithms are good enough, the ability of QICA-CARP to find solutions is not reflected completely.

Table 5 The comparison results of QICA-CARP and RDG-MAENS on *val*

name	V	T	E	LB	MAENS			RDG-MAENS			QICA-CARP			<i>p</i>	<i>h</i>	$\Delta(\%)$
					mean	S_D	Best	mean	S_D	Best	mean	S_D	Best			
1A	24	39	39	173	173.0	0.0	173	173.0	0.0	173	173.0	0.0	173	1	0	0
1B	24	39	39	173	173.0	0.0	173	179.7	1.0	179	178.0	4.2	173	0.0077	1	0
1C	24	39	39	245	245.0	0.0	245	245.1	0.7	245	245.5	2.1	245	0.5000	0	0
2A	24	34	34	227	227.0	0.0	227	239.4	8.1	227	237.9	6.8	227	0.0303	1	0
2B	24	34	34	259	259.0	0.0	259	262.5	4.5	259	261.5	2.7	259	0.0322	1	0
2C	24	34	34	457	457.2	1.1	457	465.5	5.5	457	459.3	3.6	457	8.5633e-006	1	0
3A	24	35	35	81	81.0	0.0	81	83.3	1.7	82	83.4	1.7	81	0.7240	0	1.25
3B	24	35	35	87	87.0	0.0	87	89.3	1.6	87	89.8	2.0	87	0.0056	1	0
3C	24	35	35	138	138.0	0.0	138	139.1	1.0	138	138.8	0.9	138	0.4407	0	0
4A	41	69	69	400	400.0	0.0	400	410.4	7.1	400	403.4	4.7	400	2.3085e-006	1	0
4B	41	69	69	412	412.0	0.0	412	418.1	6.1	412	414.1	4.4	412	1.7567e-005	1	0
4C	41	69	69	428	431.1	3.1	428	442.6	3.8	434	434.1	5.8	428	1.5912e-006	1	1.38
4D	41	69	69	526	532.9	3.3	530	545.4	5.0	536	533.6	3.4	530	1.5650e-006	1	1.12
5A	34	65	65	423	423.0	0.0	423	428.2	4.3	423	425.3	5.1	423	0.0010	1	0
5B	34	65	65	446	446.0	0.0	446	447.1	2.5	446	446.6	1.2	446	0.0313	1	0
5C	34	65	65	473	474.0	0.0	474	477.2	3.9	474	474.4	1.1	474	2.6963e-004	1	0
5D	34	65	65	573	582.9	2.2	577	596.4	5.7	579	583.5	3.7	577	1.0996e-006	1	0.35
6A	31	50	50	223	223.0	0.0	223	225.6	2.6	223	224.2	2.5	223	0.0042	1	0
6B	31	50	50	233	233.0	0.0	233	236.5	3.1	233	234.3	2.1	233	6.2042e-005	1	0
6C	31	50	50	317	317.0	0.0	317	319.0	2.6	317	318.3	1.9	317	0.0039	1	0
7A	40	66	66	279	279.0	0.0	279	282.5	2.8	279	280.5	4.4	279	6.8189e-004	1	0
7B	40	66	66	283	283.0	0.0	283	283.6	2.0	283	285.4	5.0	283	0.0313	1	0.84
7C	40	66	66	334	334.0	0.0	334	337.6	3.4	334	335.2	3.0	334	7.5568e-005	1	0
8A	30	63	63	386	386.0	0.0	386	391.3	6.8	386	390.3	6.0	386	0.0121	1	0
8B	30	63	63	395	395.0	0.0	395	400.9	4.8	395	396.7	3.1	395	6.9222e-005	1	0
8C	30	63	63	518	525.9	1.7	521	539.2	4.4	527	527.8	4.2	521	1.5273e-006	1	1.14
9A	50	92	92	323	323.0	0.0	323	330.0	4.6	325	325.9	2.6	323	1.5831e-006	1	0.62
9B	50	92	92	326	326.0	0.0	326	331.7	3.6	326	327.3	2.4	326	1.1594e-005	1	0
9C	50	92	92	332	332.0	0.0	332	336.3	2.9	332	333.2	2.5	332	7.0048e-006	1	0
9D	50	92	92	385	391.0	0.0	391	397.3	3.9	391	391.8	1.8	391	7.3698e-006	1	0
10A	50	97	97	428	428.0	0.0	428	436.6	4.9	430	431.1	2.8	428	1.5460e-006	1	0.47
10B	50	97	97	436	436.0	0.0	436	440.8	3.5	437	437.0	1.1	436	1.5460e-006	1	0.23
10C	50	97	97	446	446.0	0.0	446	451.2	4.1	447	447.3	1.3	446	1.2868e-006	1	0.00
10D	50	97	97	525	533.6	1.5	531	540.6	3.5	535	531.3	2.4	527	1.4392e-006	1	0.01

Table 5 shows the simulation of the results obtained by applying QICA-CARP, RDG-MAENS and MAENS on the *val* test sets. QICA-CARP and MAENS algorithms achieve the same optimal solution on all instances except 10D, on

which QICA-CARP got a better solution. On the other hand, the average obtained by MAENS is better than the one obtained by QICA, because *val* has a very small scale and the ability of global searching is weakened after using

decomposition in QICA-CARP. QICA-CARP results in 30 better solutions than RDG-MAENS whereas only four solutions obtained by RDG-MAENS and QICA-CARP are equal in *val* dataset. In addition, the S_D of QICA-CARP indicates that it results in better stability property on the *val* test set. Although, QICA-CARP converges to a better lower bound, it's not the best lower bound of the instances. According to the test results of Wilcoxon signed rank test on this instances, there are 30 cases of ' $h=1$ ' in the 34 instances which show that the new algorithm outperforms RDG-MAENS. This illustrate that the new algorithm can be used instead of the RDG-MAENS. Fig. 7 shows the convergence curves on some instances of the *val* test set. The blue line represents the result of RDG-MAENS whereas the red line is QICA-CARP.

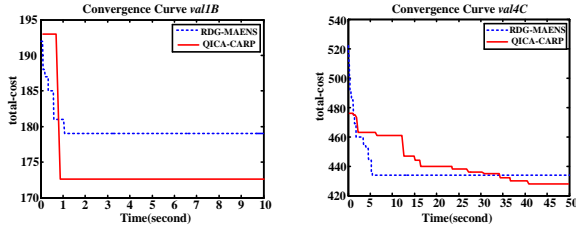


Figure 7 The convergence curves of QICA-CARP on some instances of *val*

Fig. 7 shows that QICA-CARP converges faster to the lower bound of instances while it is stable. However, the convergence rate is not ideal at start on some test set. Because the *val* is the medium-scale test set, the convergence graphs illustrate that convergence speed of QICA-CARP must be still improved. Nonetheless, the global search capability of QICA-CARP has been reflected on the medium-scale instance. We show the box diagram on the instance in Fig. 8 to analyze the statistical properties of QICA-CARP and RDG-MAENS on the test set *val*. Fig. 8 shows the box diagram about 'total_cost' of QICA-CARP and RDG-MAENS on part of instances of the *val* test set.

By analyzing the distribution of the test results in the box diagram, we see that the QICA-CARP is more stable and capable of finding the lower bound than RDG-MAENS. In addition, the number of abnormal values in the test results of

QICA-CARP is significantly less than the RDG-MAENS because of the repair operator in QICA-CARP. The results obtained for this test set demonstrate the superiority of the QICA-CARP where it can be used instead of RDG-MAENS. This is in accordance with the results gained by testing on *val*, shown in Table 5. In summary, the performance of QICA-CARP is much better than RDG-MAENS for solving the medium-scale CARP.

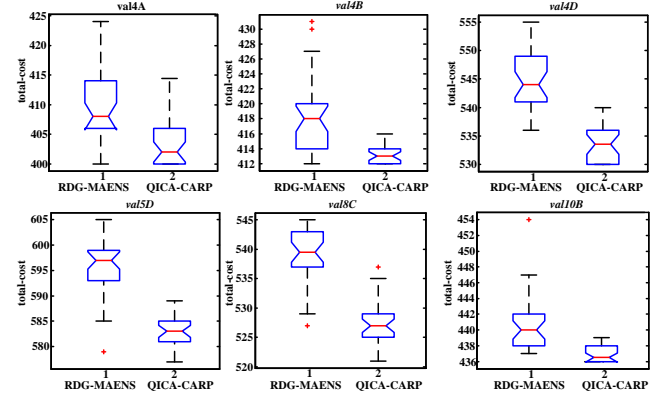


Figure 8 The box diagram of QICA-CARP and RDG-MAENS on *val* test set

Table 6 presents the results obtained by QICA-CARP, RDG-MAENS and MAENS on testing set *Beullens' C, E*. We see that QICA-CARP obtains the optimal solution on 37 out of 52 instances, whereas MAENS achieves 1 optimal solution. Moreover, MAENS results in 14 better average values, whereas QICA-CARP has 33 better average value. Compare the QICA-CARP with RDG-MAENS, 2 better solutions are obtained by using QICA-CARP out of 52 samples whereas RDG-MAENS has none. However, the improvement of QICA-CARP is not sufficient which is in accordance with the results gained by Wilcoxon signed rank test on *Beullens' C, E*. In Wilcoxon signed rank test, there are 12 ' $h=1$ ' in *Beullens' C* and 13 ' $h=1$ ' in *Beullens' E*. Besides, by analyzing the value of S_D , we can see that the stability of QICA-CARP is stronger.

Table 6 The comparison results of QICA-CARP and RDG-MAENS on *Beullen' C, E*

3	V	T	E	LB	MAENS			RDG-MAENS			QICA-CARP			p	h	$\Delta(\%)$
					mean	S_D	Best	mean	S_D	Best	mean	S_D	Best			
C01	69	98	79	4105	4162.0	23.9	4115	4172.3	27.89	4150	4172.0	25.78	4150	0.2500	0	0
C02	48	66	53	3135	3135.7	3.7	3135	3148.0	30.2	3135	3148.0	24.53	3135	0.1250	0	0
C03	46	64	51	2575	2627.8	3.9	2625	2585.3	4.07	2575	2585.0	4.5	2575	0.9047	0	0
C04	60	84	72	3478	3535.7	4.5	3533	3528.2	46.34	3510	3519.3	34.92	3510	0.0313	1	0
C05	56	79	65	5365	5477.3	30.0	5425	5415.0	27.45	5365	5415.7	32.63	5365	0.5000	0	0
C06	38	55	51	2535	2587.5	3.4	2575	2543.7	15.86	2535	2538.0	6.3	2535	0.2656	0	0
C07	54	70	52	4075	4135.0	0.0	4135	4095.0	24.49	4075	4099.3	24.59	4075	9.7742e-004	1	0
C08	66	88	63	4090	4162.3	4.3	4160	4097.5	29.52	4090	4117.3	66.85	4090	0.0050	1	0
C09	76	117	97	5233	5300.8	21.5	5278	5292.5	24.32	5260	5282.2	22.68	5260	0.5625	0	0
C10	60	82	55	4700	4827.8	43.8	4780	4755.0	43.93	4700	4749.5	36.13	4700	0.0625	0	0
C11	83	118	94	4583	4711.7	34.1	4673	4671.3	34.59	4640	4688.2	37.4	4640	0.0488	1	0
C12	62	88	72	4209	4309.0	0.0	4309	4256.3	34.97	4240	4247.3	27.56	4240	0.1563	0	0
C13	40	60	52	2955	3027.0	21.4	3015	2959.0	13.75	2955	2956.0	2.0	2955	0.0313	1	0
C14	58	79	57	4030	4097.3	10.8	4090	4036.3	10.16	4030	4032.5	7.48	4030	0.0313	1	0
C15	97	140	107	4912	5042.5	16.3	5007	4967.2	20.15	4940	4965.7	18.71	4940	0.1875	0	0
C16	32	42	32	1475	1480.2	0.9	1480	1480.7	8.34	1475	1477.8	6.54	1475	0.3008	0	0
C17	43	56	42	3555	3583.3	17.8	3575	3583.8	38.25	3555	3567.8	26.54	3555	1	0	0

C18	93	133	121	5577	5673.7	18.9	5652	5648.3	23.5	5620	5638.5	12.32	5620	3.8497e-004	1	0
C19	62	84	61	3096	3186.7	19.1	3156	3147.5	25.09	3115	3147.5	22.4	3120	4.8828e-004	1	0
C20	45	64	53	2120	2123.5	6.7	2120	2131.7	13.44	2120	2132.7	12.89	2120	0.0727	0	0
C21	60	84	76	3960	3980.2	0.9	3980	4000.0	45.28	3970	3980.2	27.61	3970	0.0430	1	0
C22	56	76	43	2245	2245.0	0.0	2245	2246.0	5.39	2245	2246.0	0.0	2245	0.0150	1	0
C23	78	109	92	4032	4136.3	30.8	4102	4102.3	24.38	4085	4098.2	20.21	4085	4.8828e-004	1	0
C24	77	115	84	3384	3422.5	6.2	3414	3412.0	9.97	3400	3407.8	8.43	3400	0.0313	1	0
C25	37	50	38	2310	2312.0	7.6	2310	2322.5	18.2	2310	2328.2	22.97	2310	1	0	0
E01	73	105	85	4885	4997.8	35.0	4965	4922.0	11.8	4910	4922.2	11.87	4910	0.0125	1	0
E02	58	81	58	3990	4025.5	14.2	4020	4004.0	27.85	3990	3998.8	19.48	3990	0.0039	1	0
E03	46	61	47	2015	2017.0	4.1	2015	2022.7	10.23	2015	2022.0	14.54	2015	0.0020	1	0
E04	70	99	77	4155	4259.3	21.1	4185	4227.0	44.13	4155	4204.0	46.34	4155	1	0	0
E05	68	94	61	4585	4683.7	49.0	4615	4649.3	71.69	4585	4640.5	49.32	4585	0.5000	0	0
E06	49	66	43	2055	2055.0	0.0	2055	2055.0	0	2055	2055.0	0	2055	0.5000	0	0
E07	73	94	50	4155	4275.0	0.0	4275	4160.3	19.96	4155	4165.7	21.51	4155	0.0176	1	0
E08	74	98	59	4710	4780.5	1.5	4780	4714.0	9.97	4710	4712.3	7.27	4710	0.0156	1	0
E09	91	141	103	5780	5947.7	38.2	5855	5899.3	48.66	5810	5889.8	51.8	5810	0.6279	0	0
E10	56	76	49	3605	3606.5	5.7	3605	3613.0	12.49	3605	3620.7	30.5	3605	0.5000	0	0
E11	80	113	94	4637	4759	44.5	4677	4723.8	42.97	4670	4732.5	42.48	4670	0.9063	0	0
E12	74	103	67	4180	4264.3	17.9	4210	4246.7	20.22	4200	4243.2	23.72	4195	0.3885	0	0.12
E13	49	73	52	3345	3385.3	22.2	3370	3349.0	14.97	3345	3348.0	0.0	3345	0.0423	1	0
E14	53	72	55	4115	4152.0	10.9	4145	4132.5	19.01	4115	4137.2	32.96	4115	1	0	0
E15	85	126	107	4189	4231.8	13.0	4229	4232.3	8.24	4220	4226.7	9.14	4215	1.4904e-006	1	0.12
E16	60	80	54	3755	3795.0	0.0	3795	3795.8	29.61	3775	3801.8	40.2	3775	0.0425	1	0
E17	38	50	36	2740	2744.3	6.3	2740	2745.0	9.83	2740	2745.7	13.52	2740	1	0	0
E18	78	110	88	3825	3837.3	6.3	3835	3856.8	26.47	3835	3856.3	27.72	3835	1.4713e-006	1	0
E19	77	103	66	3222	3259.0	3.1	3257	3251.2	25.87	3235	3251.2	23.11	3235	1	0	0
E20	56	80	63	2802	2842.0	0.0	2842	2826.3	4.99	2825	2842.3	28.14	2825	0.7734	0	0
E21	57	82	72	3728	3783.5	22.9	3733	3790.3	38.58	3730	3791.0	32.47	3730	1.5381e-006	1	0
E22	54	73	44	2470	2502.5	6.3	2500	2489.2	42.8	2470	2475.0	18.9	2470	0.4063	0	0
E23	93	130	89	3686	3760.0	13.1	3726	3752.3	38.64	3710	3743.7	32.79	3710	1.6157e-006	1	0
E24	97	142	86	4001	4129.2	26.3	4091	4062.5	29.69	4020	4059.0	28.98	4020	1.5117e-006	1	0
E25	26	35	28	1615	1615.0	0.0	1615	1663.0	67.2	1615	1661.7	63.24	1615	1.6594e-006	1	0

In Table 7, we report the results of using QICA-CARP, RDG-MAENS and MAENS to find the optimum of *Beullen'D*, *F*, which is a medium-scale test set. The reported results show QICA-CARP obtains the optimal solution on 13 out of 52 instances whereas MAENS could find only 4 optimal solutions. This indicates that QICA-CARP is more appropriate than MAENS for searching optimum of the medium scale data set because *Beullens'* decomposition strategy results in a better local searching and consequently in a better solution. Nonetheless, MAENS achieved 42 better average values whereas QICA achieved 7 because the scale of problems is not large enough. Furthermore, QICA-CARP attained 22 better solutions than RDG-MAENS out of 25

samples of *Beullens'D*. As for other 25 samples of *Beullens'F*, QICA-CARP produced 19 solutions better than the ones obtained by the RDG-MAENS. Moreover, QICA-CARP results in more stable solutions while it converges to the lower bound of all instances in this test set. Nonetheless, the new algorithm can obtain two better lower bounds in *Beullen'D*, *F*. There are 19 instances of $h=1$ in *Beullens'D* and 15 in *Beullens'F* using Wilcoxon signed rank test. The instance of *Beullens'D*, *F* are from the same graph but the capacity of *Beullens'F* is twice the *Beullens'D*. This demonstrates that QICA-CARP outperforms other algorithm in finding optimum of a large-scale test set.

Table 7 The comparison results of QICA-CARP and RDG-MAENS on *Beullen'D*, *F*

name	V	T	E	LB	MAENS			RDG-MAENS			QICA-CARP			p	h	$\Delta(\%)$
					mean	S_D	Best	mean	S_D	Best	mean	S_D	Best			
D01	69	98	79	3215	3235	0.0	3235	3242.0	1.9	3215	3240.8	1.0	3215	0.6250	0	0
D02	48	66	53	2520	2520.0	0.0	2520	2537.2	31.1	2520	2528.0	3.7	2520	9.7656e-004	1	0
D03	46	64	51	2065	2065.2	0.9	2065	2080.3	27.0	2065	2077.7	11.9	2065	0.3438	0	0
D04	60	84	72	2785	2786.0	3.8	2785	2785.0	0.0	2785	2785.0	0.0	2785	1	0	0
D05	56	79	65	3935	3935.0	0.0	3935	3945.0	23.6	3935	3950.3	20.2	3935	0.1250	0	0
D06	38	55	51	2125	2133.0	15.6	2125	2170.3	19.1	2125	2175.7	25.8	2125	0.5276	0	0
D07	54	70	52	3078	3190.3	17.7	3178	3171.3	26.5	3125	3170.0	13.3	3125	0.5762	0	0
D08	66	88	63	2995	3084.2	15.5	3065	3079.2	16.5	3045	3077.7	17.6	3045	0.3125	0	0
D09	76	117	97	4120	4120.0	0.0	4120	4126.3	10.7	4120	4131.2	5.0	4120	0.8438	0	0
D10	60	82	55	3335	3345.0	0.0	3345	3349.0	20.2	3340	3347.3	13.6	3330*	0.5000	0	0.30
D11	83	118	94	3745	3760.5	6.2	3745	3767.2	6.8	3745	3760.3	6.6	3715*	1.6184e-004	1	0.80
D12	62	88	72	3310	3310.0	0.0	3310	3329.2	32.0	3310	3323.3	21.9	3310	0.0156	1	0
D13	40	60	52	2535	2536.0	2.0	2535	2540.2	3.5	2535	2537.5	2.8	2535	0.1094	0	0
D14	58	79	57	3272	3283.0	3.1	3282	3290.0	17.9	3280	3289.0	13.6	3272	0.8750	0	0.24
D15	97	140	107	3990	3999.0	3.1	3990	4011.0	12.9	3990	4004.7	8.0	3990	0.0078	1	0
D16	32	42	32	1060	1060.0	0.0	1060	1272.0	12.3	1260	1271.2	11.3	1260	0.5000	0	0
D17	43	56	42	2620	2620.0	0.0	2620	2626.3	3.8	2620	2622.3	0.0	2620	1	0	0
D18	93	133	121	4165	4169.2	8.7	4165	4189.0	23.5	4165	4183.3	21.1	4165	0.0107	1	0
D19	62	84	61	2393	2423.0	0.0	2423	2406.2	8.9	2400	2404.7	4.0	2393	0.4609	0	0.29
D20	45	64	53	1870	1870.2	0.9	1870	1944.0	82.0	1870	1914.0	48.4	1870	3.9169e-005	1	0
D21	60	84	76	2985	3024.2	18.8	2995	3098.8	21.1	3055	3092.5	15.3	3045	0.0234	1	0.00
D22	56	76	43	1865	1865.0	0.0	1865	1932.3	88.7	1865	1940.8	74.9	1865	0.4952	0	0
D23	78	109	92	3114	3147.2	8.4	3134	3158.2	13.8	3130	3155.3	16.2	3130	0.0167	1	0
D24	77	115	84	2676	2739.5	19.3	2726	2740.3	22.9	2710	2734.7	22.8	2710	0.0664	0	0

D25	37	50	38	1815	1815.0	0.0	1815	1840.3	44.2	1815	1819.0	7.4	1815	0.0078	1	0
F01	73	105	85	4040	4046.0	7.0	4040	4060.5	18.2	4040	4049.2	4.1	4040	3.5768e-005	1	0
F02	58	81	58	3300	3300.0	0.0	3300	3311.3	15.7	3300	3313.2	10.5	3300	0.6250	0	0
F03	46	61	47	1665	1665.0	0.0	1665	1696.2	36.0	1665	1682.5	13.4	1665	0.0020	1	0
F04	70	99	77	3476	3498.2	8.4	3476	3507.5	9.6	3485	3499.8	7.9	3435*	8.3218e-004	1	1.43
F05	68	94	61	3605	3605.3	1.3	3605	3605.3	1.3	3605	3603.7	2.0	3600*	0.0020	0	0.14
F06	49	66	43	1875	1875.0	0.0	1875	1909.8	39.2	1875	1890.2	17.6	1875	9.7656e-004	1	0
F07	73	94	50	3335	3345.7	24.5	3335	3393.5	62.5	3335	3366.5	34.8	3335	6.1035e-005	1	0
F08	74	98	59	3690	3700.0	0.0	3700	3712.3	11.0	3705	3715.7	12.2	3705	0.1250	0	0
F09	91	141	103	4730	4782.8	27.4	4730	4801.8	36.0	4730	4798.8	39.6	4730	0.1546	0	0
F10	56	76	49	2925	2925.0	0.0	2925	2936.5	4.9	2925	2934.2	4.9	2925	1	0	0
F11	80	113	94	3835	3857.5	16.3	3835	3864.8	19.5	3835	3853.8	13.5	3835	8.2229e-005	1	0
F12	74	103	67	3390	3429.5	33.0	3400	3462.3	35.2	3395	3450.0	26.3	3395	1.0036e-004	1	0
F13	49	73	52	2855	2855.0	0.0	2855	2860.0	7.1	2855	2858.5	4.6	2855	0.3125	0	0
F14	53	72	55	3330	3370.5	14.4	3330	3398.8	25.7	3330	3384.2	12.1	3330	5.2563e-005	1	0
F15	85	126	107	3560	3566.7	9.9	3560	3575.7	13.5	3560	3573.7	10.4	3560	0.0410	1	0
F16	60	80	54	2725	2725.0	0.0	2725	2741.7	25.6	2725	2766.8	28.7	2725	0.0156	1	0
F17	38	50	36	2055	2055.0	0.0	2055	2058.0	0.0	2055	2055.0	0.0	2055	1	0	0
F18	78	110	88	3063	3089.2	23.3	3078	3126.3	52.6	3075	3123.0	43.1	3065	0.4833	0	0.33
F19	77	103	66	2500	2540	0.0	2540	2527.0	2.0	2525	2525.0	0.0	2525	0.2500	0	0
F20	56	80	63	2445	2449.8	0.9	2445	2450.3	2.8	2445	2451.7	2.7	2445	0.1250	0	0
F21	57	82	72	2930	2930.0	0.0	2930	2943.3	18.3	2930	2933.0	1.5	2930	0.0039	1	0
F22	54	73	44	2075	2075.0	0.0	2075	2077.8	5.0	2075	2081.7	5.5	2075	0.0156	1	0
F23	93	130	89	2994	3124.3	14.3	3014	3015.7	9.6	3005	3013.7	5.7	3000	0.0088	1	0.17
F24	97	142	86	3210	3236.3	17.1	3210	3253.5	12.6	3220	3249.2	10.4	3210	6.1035e-005	1	0.31
F25	26	35	28	1390	1390.0	0.0	1390	1510.2	69.2	1390	1519.0	61.1	1390	0.0020	1	0

In summary, the results illustrate that QICA-CARP is superior to RDG-MAENS both the stability and the searching capability of these four test sets including *Beullen'*C, D, E, F. In addition, Fig. 9 shows convergence behavior of QICA-CARP during computing the optimum of these test set.

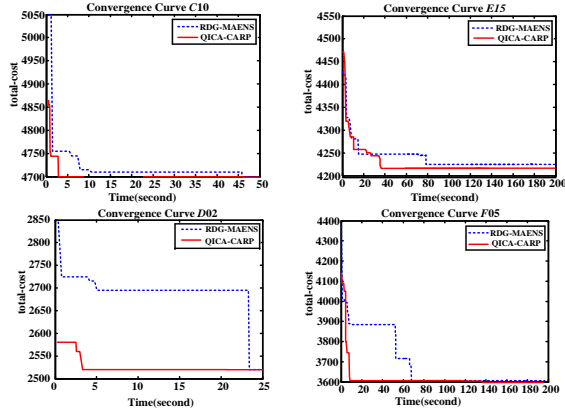


Figure 9 The convergence curves of QICA-CARP on some instances of *Beullen'*C, D, E, F

Fig. 9 shows that QICA-CARP finds a better solution for E15 and F05 test set whereas the other two algorithms obtain the same results. Nonetheless, QICA-CARP has a faster convergence indicating that the convergence speed and stability of QICA-CARP are better than the RDG-MAENS. We also plot the box diagram of some examples of the test sets to better illustrate the statistical results of these two algorithms in finding the optimum of *Beullen'*C, D, E, F test sets. In specific, Fig. 10 shows the box diagram of 'total_cost' showing that QICA-CARP results in a better solution than RDG-MAENS and shows better stability behavior on some instances.

The number of the abnormal values obtained by QICA-CARP is significantly less than the one obtained by RDG-MAENS. This indicates that the QICA-CARP is more capable of repairing the infeasible solution than

RDG-MAENS. Fig. 10 also shows the superiority of the QICA-CARP to the RDG-MAENS in terms of the maximum value, the median line, the minimum value and the emphasis of data distribution for solving CARP problem.

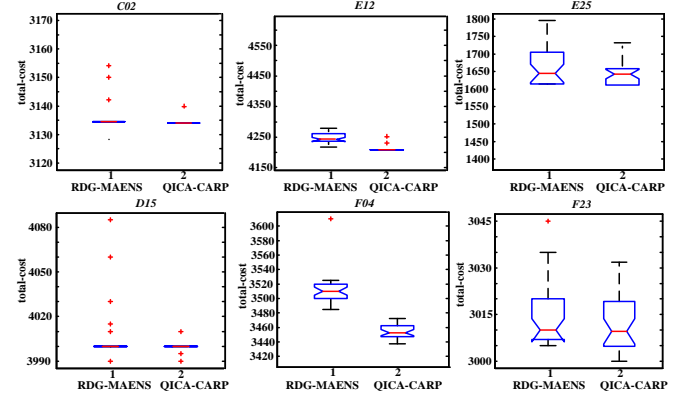


Figure 10 The box diagram of QICA-CARP and RDG-MAENS on *Beullen'*C, D, E, F test set

In Table 8, we report the test results obtained by QICA-CARP, RDG-MAENS and MAENS for solving the large-scale test set *egl*. It is evident that QICA-CARP obtained a better optimal solution for 10 out of 24 instances whereas MAENS obtained 3 better optimal solutions. Furthermore, MAENS resulted in 8 better average values where QICA has 13 better average values. The reason is that *egl* is a large scale test set and QICA-CARP used decomposition strategy to improves the ability of local searching and make full use of the computing sources. In comparison with RDG-MAENS, QICA-CARP obtained 22 better solutions out of 24 instances of the *egl* test set. This is the average total cost of the best-so-far solutions over the 30 independent runs. The S_D values show that QICA-CARP results in more stable solutions than RDG-MAENS. Nonetheless, QICA-CARP found optimal solutions for 10 test instances better than the one obtained by RDG-MAENS. The

Wilcoxon signed rank test analysis results illustrate that there are 17 instances with $h=1$ verifying the superiority of QICA-CARP to RDG-MAENS. Moreover, QICA-CARP

achieved better solution than RDG-MAENS on the large-scale *egl* instances.

Table 8 The comparison results of QICA-CARP and RDG-MAENS on *egl*

name	V	T	E	LB	MAENS			RDG-MAENS			QICA-CARP			p	h	$\Delta(\%)$
					mean	S_D	Best	mean	S_D	Best	mean	S_D	Best			
E1-A	77	51	98	3548	3548.0	0.0	3548	3556.7	28.0	3548	3554.1	19.1	3548	0.3750	0	0
E1-B	77	51	98	4498	4516.5	17.6	4498	4530.7	16.3	4498	4520.1	15.9	4498	1.2474e-004	1	0
E1-C	77	51	98	5595	5601.6	9.9	5595	5621.4	26.1	5595	5609.3	25.5	5595	1.7199e-005	1	0
E2-A	77	72	98	5018	5018.0	0.0	5018	5026.8	32.2	5018	5018.3	1.6	5018	0.1250	0	0
E2-B	77	72	98	6305	6341.4	12.0	6317	6344.7	19.6	6317	6341.7	20.6	6314	0.0012	1	0.05
E2-C	77	72	98	8335	8355.7	35.9	8335	8358.1	41.0	8335	8352.6	28.8	8335	0.0625	0	0
E3-A	77	87	98	5898	5898.8	2.9	5898	5913.5	31.1	5898	5917.3	34.8	5898	0.0781	0	0
E3-B	77	87	98	7729	7802.9	27.3	7775	7817.8	29.9	7777	7800.4	20.8	7775	2.4864e-006	1	0.03
E3-C	77	87	98	10244	10321.9	18.0	10292	10327.9	30.1	10292	10321.4	21.5	10292	0.0013	1	0
E4-A	77	98	98	6408	6475.2	10.3	6456	6479.8	18.6	6461	6476.2	25.6	6446	0.0040	1	0
E4-B	77	98	98	8935	9023.0	18.7	8998	9028.4	23.8	8975	9022.5	25.8	8975	2.1732e-004	1	0
E4-C	77	98	98	11493	11645.8	46.7	11561	11654.5	58.3	11594	11628.8	37.6	11567	3.7732e-006	1	0.03
S1-A	140	75	190	5018	5039.8	35.9	5018	5059.5	44.8	5018	5073.3	40.6	5018	0.1145	0	0
S1-B	140	75	190	6388	6433.4	8.6	6388	6424.5	29.7	6388	6415.3	21.1	6388	0.0117	1	0
S1-C	140	75	190	8518	8518.3	1.5	8518	8541.9	30.7	8518	8531.7	22.4	8518	1.2674e-004	1	0
S2-A	140	147	190	9825	9959.2	34.6	9895	10000.9	65.3	9909	9976.3	43.3	9909	4.3218e-005	1	0
S2-B	140	147	190	13017	13231.6	63.2	13147	13203.5	70.3	13124	13185.0	68.0	13110	0.1629	0	0.06
S2-C	140	147	190	16425	16509.8	51.8	16430	16488.6	43.7	16425	16474.7	39.2	16425	7.7653e-006	1	0
S3-A	140	159	190	10146	10312.7	26.5	10257	10288.5	43.5	10242	10286.8	44.4	10221	0.3540	0	0.20
S3-B	140	159	190	13648	13876.6	67.8	13749	13814.1	70.2	13715	13803.4	63.3	13692	0.0314	1	0.13
S3-C	140	159	190	17188	17305.8	41.4	17207	17288.7	38.5	17216	17271.6	36.4	17214	3.8571e-006	1	0.01
S4-A	140	190	190	12144	12419.2	33.2	12341	12388.6	54.6	12293	12380.1	56.8	12265	0.0094	1	0.23
S4-B	140	190	190	16103	16441.2	38.1	16337	16407.7	59.3	16262	16387.2	57.9	16262	7.9366e-006	1	0
S4-C	140	190	190	20430	20767.2	74.6	20538	20672.1	73.7	20530	20689.8	87.5	20505	4.0126e-004	1	0.12

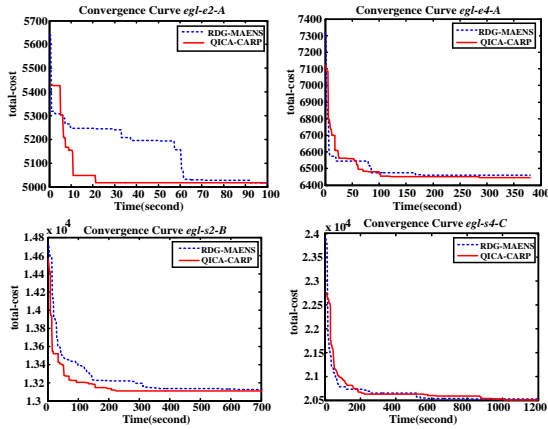


Figure 11 The convergence curves of QICA-CARP on some instances of *egl*

Figure 11 gives a more detailed description of the test results of QICA-CARP and RDG-MAENS on the *egl* test set. According to the results reported in Table 8, QICA-CARP found a better average value than RDG-MAENS on some instances using the same parameters. Fig. 11 shows that the convergence rate of QICA-CARP is slower than RDG-MAENS at the initial stage for some test instances. Nonetheless, QICA-CARP found a better solution faster than the one obtained by RDG-MAENS where the solution is more stable than the one obtained by RDG-MAENS.

Fig. 12 shows the box diagram about ‘total_cost’ of QICA-CARP and RDG-MAENS on the *egl* test set. It is seen that QICA-CARP can get a better lower bound and a lower median line than the one obtained by RDG-MAENS. This indicates that the new algorithm results in a better minimum consumption path for solving CARP for these instances. In

addition, the probability of abnormal values computed by RDG-MAENS is bigger than the one computed by QICA-CARP. This illustrates that the repairing of the infeasible solutions in QICA-CARP is very effective. In summary, QICA-CARP can find a better solution on the instance in figure 12 than the ones obtained by RDG-MAENS on large-scale test set *egl*.

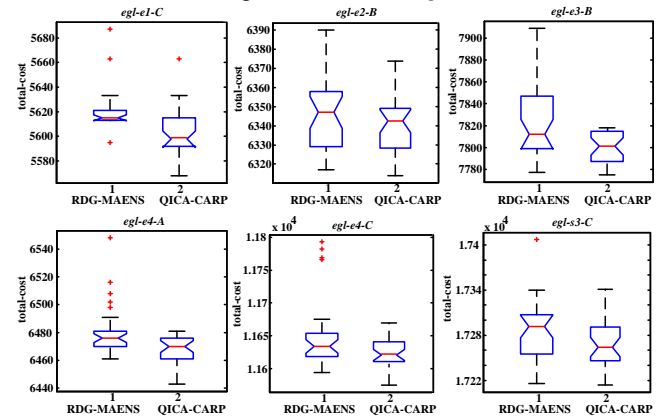


Figure 12 The box diagram of QICA-CARP and RDG-MAENS on *egl* test set

To test the performance more completely, we added the RMA algorithm [35] to make a comparison. Table 9 shows the results obtained for the test set *EGL-G* after 30 runs. It also shows the average solutions of ‘total-cost’ obtained by RDG-MAENS, QICA-CARP and RMA. The results show QICA-CARP can find 8 better solutions on 10 instances of the test set *EGL-G* indicating that QICA-CARP improves significantly the test outcome for the problem with the very large-scale test set of *EGL-G*. However, the standard

deviation of the results obtained by QICA-CARP is large indicating that we need to improve the stability of QICA-CARP on these instances. In addition, QICA-CARP converges to a better lower bound on 7 instances. The results reported in Table 9 also illustrate the capability of global search of QICA-CARP for searching a large-scale test set. Furthermore, QICA-CARP obtained 7 instances with $h=1$ out

of 10 instances according to the Wilcoxon signed rank test. QICA-CARP got 6 solutions better than RMQ where RMA got 4. On the other hand, QICA obtained 1 better average where RMA got 9. We can see that QICA has a slightly better performance in searching the best solution, but it is not competitive in average value compared with RMA.

Table 9 The comparison results of QICA-CARP and RDG-MAENS on *EGL-G*

name	V	T	E	LB	RMA		RDG-MAENS			QICA-CARP			p	h	$\Delta(\%)$
					mean	Best	mean	S_D	Best	mean	S_D	Best			
G1-A	255	375	347	1001210	1007693.7	996675	1008717.5	4670.4	1001210	1008151.8	4441.1	999151*	5.7064e-04	1	0.20
G1-B	255	375	347	1118596	1054639.2	1002079	1126652.7	5312.7	1118596	1125874.0	5802.8	1118030*	7.1570e-04	1	0.05
G1-C	255	375	347	1245398	1247471.667	1246507	1254743.4	4704.2	1245398	1252912.8	5242.1	1245398	4.1725e-05	1	0
G1-D	255	375	347	1380711	1375893.5	1375289	1388719.2	5334.1	1380711	1387461.7	6012.5	1376795*	0.4779	0	0.03
G1-E	255	375	347	1521171	1530297	1524372	1533089.5	8133.8	1521171	1529252.2	6101.5	1518055*	0.0207	1	0.20
G2-A	255	375	375	1101797	1107473.818	1101465	1108472.7	3559.4	1101797	1109462.4	5923.1	1100447*	0.1359	0	0.12
G2-B	255	375	375	1213093	1217433	1213383	1223670.2	5742.1	1213093	1222531.7	4843.3	1213004*	1.3034e-04	1	0.00
G2-C	255	375	375	1342537	1352476.5	1351292	1354538.8	5551.1	1342537	1356537.0	5344.8	1344221	7.5137e-05	1	0
G2-D	255	375	375	1486584	1486958.5	1482597	1493660.2	4465.0	1486584	1492428.0	3696.0	1482861*	1.4773e-04	1	0.25
G2-E	255	375	375	1624438	1633968	1633026	1637388.9	7736.6	1624438	1636746.5	5764.3	1625984	0.5440	0	0

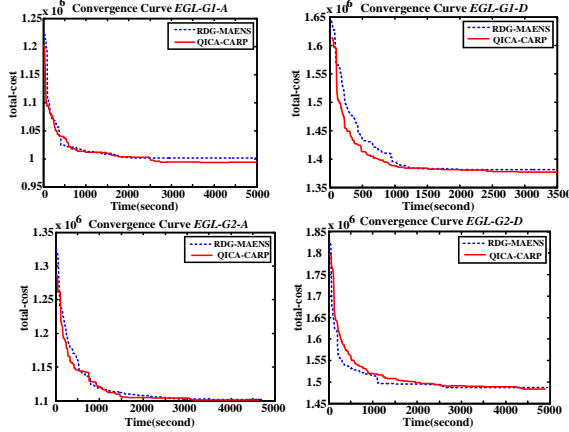


Figure 13 The convergence curves of QICA-CARP on *EGL-G*

According to the results reported in Table 9, QICA-CARP obtained better solutions than RDG-MAENS for the test set *EGL-G*. We also show the convergence of the two algorithms in Fig. 13 during computation the optimal solution of the test set *EGL-G*. With the same parameter settings, the convergence of QICA-CARP is faster than the RDG-MAENS. Although the convergence rate of QICA-CARP is not very good at initial evolution of the algorithm, it found solutions better than RDG-MAENS. After initial stage of the algorithm evolution, the convergence rate speeds up. In general, the improvement demonstrated by QICA-CARP is beneficial to find a better solution especially for the large-scale test set.

Figure 14 shows the box diagram of the 'total_cost' obtained using QICA-CARP and RDG-MAENS to find the optimum of the test set *EGL-G*. It is clear that QICA-CARP can find a better lower bound than RDG-MAENS. Although the stability needs to be improved on some instances, QICA-CARP is superior to RDG-MAENS in terms of performance on most instances. In QICA-CARP, the data below the median line in the box diagram is relative much.

This indicates that QICA-CARP outperforms RDG-MAENS in finding the minimum consumption path for the large-scale test set. In conclusion, QICA-CARP is suitable for solving the large-scale CARP

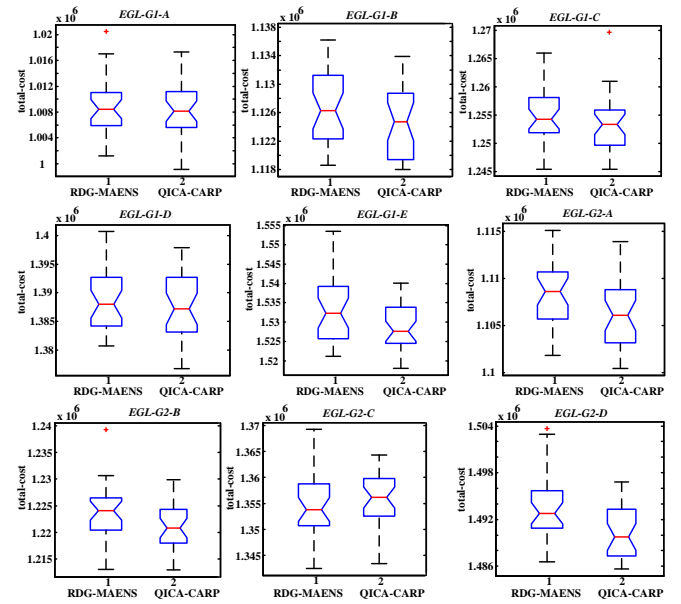


Figure 14 The box diagram of QICA-CARP and RDG-MAENS on *EGL-G* test set

C. The comparison among different versions of QICA-CARP

QICA-CARP mainly includes 2 improvements: (1) adds the mutation strategy of quantum rotation gate and (2) uses the repair operator. In order to investigate the effect of each operation, we consider three different versions of the algorithm as follows: (1) in version 1, QICA-CARP is combined mutation of quantum rotation gate, (2) in version 2, QICA-CARP is combined repair operator and (3) in version 3, QICA-CARP is combined with mutation and repair operator.

Table 10 The comparison among different versions of QICA-CARP on *egl*

name	V	T	E	LB	RDG-MAENS			1			2			3		
					mean	S_D	Best	mean	S_D	Best	mean	S_D	Best	mean	S_D	Best
E1-A	77	51	98	3548	3556.7	28.0	3548	3551.9	19.1	3548	3550.3	17.8	3548	3548.6	3.3	3548
E1-B	77	51	98	4498	4530.7	16.3	4498	4526.5	13.5	4498	4529.9	13.2	4498	4532.4	14.7	4498
E1-C	77	51	98	5595	5621.4	26.1	5595	5617.1	25.4	5595	5627.2	31.8	5595	5613.3	22.1	5595
E2-A	77	72	98	5018	5026.8	32.2	5018	5029.0	33.0	5018	5029.6	35.6	5018	5024.6	23.3	5018
E2-B	77	72	98	6305	6344.7	19.6	6317	6339.3	16.1	6317	6342.3	18.5	6317	6352.9	27.4	6317
E2-C	77	72	98	8335	8358.1	41.0	8335	8362.6	41.3	8335	8356.7	36.6	8335	8369.1	42.2	8335
E3-A	77	87	98	5898	5913.5	31.1	5898	5925.6	47.0	5898	5916.0	35.2	5898	5913.5	28.8	5898
E3-B	77	87	98	7729	7817.8	29.9	7777	7814.3	26.9	7777	7810.7	28.1	7777	7810.6	26.5	7777
E3-C	77	87	98	10244	10327.9	30.1	10292	10330.1	32.2	10292	10326.6	29.4	10292	10322.1	18.5	10292
E4-A	77	98	98	6408	6479.8	18.6	6461	6478.7	17.9	6446	6472.5	14.5	6446	6484.3	30.3	6446
E4-B	77	98	98	8935	9028.4	23.8	8975	9020.8	22.3	8962	9034.4	23.3	8988	9023.8	22.7	8962
E4-C	77	98	98	11493	11654.5	58.3	11594	11643.1	40.0	11594	11665.7	59.5	11594	11644.5	54.1	11594
S1-A	140	75	190	5018	5059.5	44.8	5018	5067.6	41.9	5018	5052.7	42.8	5018	5063.6	46.6	5018
S1-B	140	75	190	6388	6424.5	29.7	6388	6418.7	27.0	6388	6421.4	31.2	6394	6417.6	22.1	6388
S1-C	140	75	190	8518	8541.9	30.7	8518	8540.5	30.1	8518	8538.3	29.3	8518	8530.0	21.7	8518
S2-A	140	147	190	9825	10000.9	65.3	9909	9974.0	44.9	9905	9976.4	44.4	9914	9971.2	43.0	9905
S2-B	140	147	190	13017	13203.5	70.3	13124	13217.3	67.1	13111	13201.6	53.5	13121	13189.4	53.6	13121
S2-C	140	147	190	16425	16488.6	43.7	16425	16483.4	42.4	16425	16488.2	42.6	16425	16476.4	38.1	16425
S3-A	140	159	190	10146	10288.5	43.5	10242	10285.3	43.2	10220	10280.5	28.7	10221	10276.4	28.4	10221
S3-B	140	159	190	13648	13814.1	70.2	13715	13827.5	73.7	13695	13786.4	56.8	13688	13791.5	62.1	13695
S3-C	140	159	190	17188	17288.7	38.5	17216	17275.5	32.1	17191	17292.4	39.1	17189	17269.9	26.7	17191
S4-A	140	190	190	12144	12388.6	54.6	12293	12396.0	46.7	12289	12388.3	52.2	12285	12389.6	44.1	12289
S4-B	140	190	190	16103	16407.7	59.3	16262	16382.7	48.3	16243	16404.9	45.0	16290	16386.2	51.3	16243
S4-C	140	190	190	20430	20672.1	73.7	20530	20686.6	77.2	20488	20679.6	74.9	20534	20672.8	79.2	20488

Table 10 shows that the version 3 of QICA-CARP consistently outperforms the other two versions. This illustrates that the combining of two improvements is more powerful than using just one of them. Moreover, the experimental results of the version 2 and 3 are significantly better than the first one indicating that both operators are effective in improving the quality of the solution. This evidences that (1) The mutation strategy of rotation gate improves the searching ability of local search and resulting in a better solution in comparison to RDG-MAENS, (2) the repair operator improves both the quality and the stability of the solution (3) by using both simultaneously results in an overall good improved performance.

D. Summary for tables of simulation results

We make a summary of the simulation results reported in tables 4 to 10 to improve the presentation clarity in terms of the times of winners on optimal solution and performance of each algorithm.

In Table 11, we compare MAENS, RDG-MAENS and RMA with QICA-CARP, consecutively. The data in table represents the number corresponding with the one getting a better solution between two algorithms, including the same results.

Table 11. The statistic of experimental results of three algorithms

	Algorithms	gdb		val		Bullens		egl		EGL-G	
		Best	Average	Best	Average	Best	Average	Best	Average	Best	Average
1	MAENS	23	5	33	32	54	64	14	11	-	-
	QICA-CARP	23	1	34	0	64	48	21	16	-	-
2	RDG-MAENS	23	7	4	5	92	27	14	2	3	2
	QICA-CARP	23	22	34	30	103	76	24	22	8	8
3	RMA	-	-	-	-	-	-	-	-	4	8
	QICA-CARP	-	-	-	-	-	-	-	-	6	2

From Table 11, compared with MAENS, we see that MAENS slightly outperforms QICA-CARP if the scale of the problem to be solved is small because decomposition strategy is not suitable for small-scale problem. If the test set is larger scale, e.g. *egl*, QICA-CARP obtains a performance better than MAENS. As for the comparison with RDG-MAENS, because RDG-MAENS and QICA-CARP utilize the decomposition strategy, their performance on small-scale problems is not satisfactory. Nonetheless, QICA-CARP shows a significant superiority than RDG-MAENS on different type of problems.

QICA-CARP (1) applies the clone operation to the dominant population and uses the information on the current optimal antibody to control population with a high probability

evolution toward good schema by the action of quantum rotation gate and (2) uses strategy of adjusting the angle of rotation dynamically to accelerate the convergence of the original clone operator. These two are the main differences between QICA-CARP and RDG in solving sub-problems. Moreover, the quantum crossover operation is used to enhance information exchange and to increase the diversity of the population. As a result, QICA-CARP obtains a satisfactory performance in comparison to RDG-MAENS. This makes QICA-CARP more suitable for the large-scale problems. QICA-CARP slightly outperforms RMA in finding an optimum of LSCARP in terms of searching. However, it is not competitive to RMA in general. All the experimental

results showed effectiveness of QICA-CARP in solving CARP especially in large-scale problems.

V. CONCLUSIONS AND FUTURE WORK

In this paper, we used the RDG decomposition strategy of RDG-MAENS for solving CARP problem. We propose an approach called QICA-CARP combining the feature of artificial immune system and quantum computation to solve CARP based on the qubit and the quantum superposition.

The proposed approach showed a performance slightly better than RDG-MAENS with a strong stability. Nonetheless, for small and medium scale test set including *gdb*, *val* and *Beulles*, QICA-CARP did not outperform MAENS. By contrast, for large-scale test *egl*, the stability and the ability to search the solutions shows slight superiority to other approaches. Eventually, we showed that QICA-CARP converges to a better solution with a faster speed compared with other algorithms in the case of large-scale test set *EGL-G* due to the global search.

At different part of algorithm evolution, the convergence is not stable because the algorithm is sensitive to the parameter. Moreover, the stability of the average solution needs to be improved on this instance. In future research, in order to make the approach applicable to even more practical problems, we need to reduce the complexity of the algorithm and to improve its stability for solving large scale arc routing problem.

ACKNOWLEDGMENT

We would like to express our sincere appreciation to the editors and the anonymous reviewers for their insightful comments, which have greatly helped us in improving the quality of the paper.

REFERENCES

- [1] R. H. Shang, J. Wang, L. C. Jiao, Y. Y. Wang. An improved decomposition-based memetic algorithm for multi-objective capacitated arc routing problem, *Appl. Soft Comput.* 19(1) (2014) 343-361.
- [2] X. Z. Wen, L. Shao, Y. Xue, and W. Fang. A rapid learning algorithm for vehicle classification, *Information Sciences.* 295 (1) (2015) 395-406.
- [3] Y. Mei, K. Tang, X. Yao. Decomposition-based memetic algorithm for multi-objective capacitated arc routing problems, *IEEE Trans. Evol. Comput.* 15 (2) (2011) 151-165.
- [4] J. Shen, H. W. Tan, J. Wang, J. W. Wang, and S. Y. Lee. A Novel Routing Protocol Providing Good Transmission Reliability in Underwater Sensor Networks, *Journal of Internet Technology.* 16 (1) (2015) 171-178.
- [5] B. L. Golden, J. S. DeArmon, E. K. Baker. Computational experiments with algorithms for a class of routing problems, *Comput. Oper. Res.* 10 (1) (1983) 47-59.
- [6] G. Ulusoy. The fleet size and mix problem for capacitated arc routing, *Eur. J. Oper. Res.* 22 (3) (1985) 329-337.
- [7] W. L. Pearn. Augment-insert algorithms for the capacitated arc routing problem, *Comput. Oper. Res.* 18 (2) (1991) 189-198.
- [8] W. L. Pearn. Approximate solutions for the capacitated arc routing problem, *Comput. Oper. Res.* 16 (6) (1989) 589-600.
- [9] K. C. Tan, E. F. Khor, T. H. Lee. Multi-objective evolutionary algorithms and applications, Springer-Verlag, May 4, (2005).
- [10] Z. H. Xia, X. H. Wang, X. M. Sun, and Q. Wang. A Secure and Dynamic Multi-keyword Ranked Search Scheme over Encrypted Cloud Data, *IEEE Transactions on Parallel and Distributed Systems.* 27 (2) (2015) 340-352.
- [11] M. Reed, A. Yiannakou, R. Evering. An ant colony algorithm for the multi-compartment vehicle routing problem, *Appl. Soft Comput.* 15 (2014) 169-176.
- [12] F. Glover, M. Laguna. *Tabu search*, Kluwer Academic, Boston, MA, 1997.
- [13] Y. Mei, K. Tang, X. Yao. A memetic algorithm for periodic capacitated arc routing problem, *IEEE Trans. Syst. Man Cybern. B* 41 (6) (2011) 1654-1667.
- [14] Y. H. Zheng, B. Jeon, D. H. Xu, Q. M. J. Wu, and H. Zhang. Image segmentation by generalized hierarchical fuzzy C-means algorithm, *Journal of Intelligent and Fuzzy System.* 28 (2) (2015) 961-973.
- [15] M. C. Goldberg, and P. H. Asconavieta. Memetic algorithm for the Traveling Car Renter Problem: an experimental investigation, *Memetic Computing.* 4 (2) (2012) 89-108.
- [16] L. Feng, Y. S. Ong, A. H. Tan, I. W. Tsang. Memes as building blocks: a case study on evolutionary optimization + transfer learning for routing problems, *Memetic Computing.* 7 (3) (2015) 159-180.
- [17] A. Hertz, G. Laporte, M. Mittaz. A tabu search heuristic for the capacitated arc routing problem, *Oper. Res.* 48 (1) (2000) 129-135.
- [18] A. Hertz and M. Mittaz. A variable neighborhood descent algorithm for the undirected capacitated arc routing problem. *Trans. Sci.* 35(4) (2001) 425-434.
- [19] P. Lacomme, C. Prins, W. Ramdane-Cherif. Competitive memetic algorithms for arc routing problem, *Ann. Oper. Res.* 131(1-4) (2004) 159-185.
- [20] R. H. Shang, L. C. Jiao, F. Liu. A novel immune clonal algorithm for MO problems, *IEEE Trans. Evol. Comput.* 16 (1) (2012) 35-49.
- [21] M. Gong, C. Liu, L. Jiao, G. Cheng. Hybrid immune algorithm with Lamarckian local search for multi-objective optimization, *Memetic Computing.* 2 (1) (2010) 47-67.
- [22] K. Tang, Y. Mei, X. Yao. Memetic algorithm with extended neighborhood search for capacitated arc routing problems, *IEEE Trans. Evol. Comput.* 13 (5) (2009) 1151-1166.
- [23] Y. Mei, X. D. Li, X. Yao. Cooperative co-evolution with route distance grouping for large-scale capacitated arc routing problems, *IEEE Trans. Evol. Comput.* 18 (3) (2014) 435-449.
- [24] A. Tonda, E. Lutton, G. Squillero. A benchmark for cooperative coevolution, *Memetic Computing.* 4 (4) (2012) 263-277.
- [25] M. L. Nguyen, S. C. Hui, A. C. M. Fong. Divide-and-conquer memetic algorithm for online multi-objective test paper generation, *Memetic Computing.* 4 (1) (2012) 33-47.
- [26] R. H. Shang, Y. Y. Wang, J. Wang, L. C. Jiao, S. Wang, L. P. Qi. A multi-population cooperative coevolutionary algorithm for multi-objective capacitated arc routing problem, *Inform. Sci.* 27 (7) (2014) 609-642.
- [27] E. Dijkstra. A note on two problems in connexion with graphs, *Number. math.* 1 (1) (1959) 269-271.
- [28] M. Marinaki, Y. Marinakis. A hybridization of clonal selection algorithm with iterated local search and variable neighborhood search for the feature selection problem, *Memetic Computing.* 7 (3) (2015) 181-201.
- [29] L. N. De Castro, F. J. Von Zuben. Learning and optimization using the clonal selection principle, *IEEE Transactions on Evolutionary Computation.* 6 (3) (2002) 239-251.
- [30] L. C. Jiao, Y. Y. Li, M. G. Gong, and X. R. Zhang. Quantum-inspired immune clonal algorithm for global optimization, *IEEE Transactions on Systems, Man, and Cybernetics Part B (Cybernetics).* 38 (5) (2008) 1234-1253.
- [31] Y. Y. Li, L. C. Jiao. Quantum-inspired immune clonal algorithm, In *Proceedings of the 4th International Conference on Artificial Immune Systems*, Banff, Alberta, Canada, (2005) 304-317.
- [32] H. Handa, L. Chapman, X. Yao. Robust route optimization for gritting/salting trucks: A CERCIA experience, *IEEE Comput. Intell. Mag.* 1 (1) (2006) 6-9.
- [33] Y. Mei, K. Tang, X. Yao. A global repair operator for capacitated arc routing problem, *IEEE Trans. Syst. Man Cyber. B.* 39 (3) (2009) 723-734.
- [34] B. L. Golden, J. S. DeArmon, E.K. Baker. Computational experiments with algorithms for a class of routing Problems, *Comput. Oper. Res.* 10 (1) (1983) 47-59.
- [35] Z. R. Wang, H. Y. Jin, M. M. Tian. Rank-based memetic algorithm for capacitated arc routing problems, *Applied soft computing.* 37 (2015) 572-584.